

Portland State University

PDXScholar

Dissertations and Theses

Dissertations and Theses

Spring 8-1-2019

Drafting in Self-Timed Circuits

Christopher Lee Cowan
Portland State University

Follow this and additional works at: https://pdxscholar.library.pdx.edu/open_access_etds



Part of the [Electrical and Computer Engineering Commons](#)

Let us know how access to this document benefits you.

Recommended Citation

Cowan, Christopher Lee, "Drafting in Self-Timed Circuits" (2019). *Dissertations and Theses*. Paper 5099.
<https://doi.org/10.15760/etd.6975>

This Dissertation is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: pdxscholar@pdx.edu.

Drafting in Self-Timed Circuits

by

Christopher Lee Cowan

A dissertation submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
in
Electrical and Computer Engineering

Dissertation Committee:

Xiaoyu Song, Chair

Ivan Sutherland

C. Glenn Shirley

John Acken

Bryant York

Jingke Li

Portland State University
2019

© 2019 Christopher Lee Cowan

Abstract

Intervals between data items propagating in self-timed circuits are controlled by handshake signals rather than by a clock. In many self-timed designs, a trailing data item will catch up with a leading item or token, even when it trails by thousands of gate delays. This effect, called “drafting,” can be seen in many of the self-timed designs, e.g., GasP, Mousetrap, Click, and Micropipeline. The purpose of this dissertation is to reveal the circuit mechanism of drafting in self-timed circuits typically used in FIFO stages. Drafting is usually considered to be incidental to the operation of self-timed circuits since interval timing information is irrelevant to preservation of the proper order of data. However, if new applications of self-timed designs require preservation of timing between data items, or if interval data carries information, then the drafting mechanism must be understood to control it. Since drafting is an analog function in a digital circuit the effect may be used as a source of randomness or uniqueness. The drafting effect changes with manufacturing variability and each unit may provide a source for a unique digital signature that can be used in security applications.

Acknowledgements

I wish to thank Drs. Sutherland, Shirley and Daasch for their encouragement and support in the many aspects of this research. Also, many thanks to the Asynchronous Research Center which directly supported me along the way.

Table of Contents

Abstract	i
Acknowledgements	ii
List of Figures	iv
Glossary	vii
Chapter 1 Introduction and Background	1
1.1 Basic GasP Operation	3
1.2 Basic GasP Timing	7
1.3 Depictions of Drafting Behavior	9
1.4 The Conventional Explanation	12
1.5 Conventional Mitigation Approaches	16
Chapter 2 Strategy	18
2.1 The Basic GasP FIFO	18
2.2 Initial Observations	19
2.3 Substitution Study	22
2.4 Measuring V_K and t_{plh} of the NOR Gate	26
2.5 The PMOS Stack Circuitry	30
Chapter 3 Modeling of Drafting	33
3.1 Deriving a K Function	33
3.2 Behavior of the Simplified Circuit	37
3.3 The Body Effect on V_K	39
3.4 MATLAB Modeling of Drafting	41
3.5 Line of Demarcation	46
Chapter 4 A Control Circuit	51
Chapter 5 Drafting with Different Technologies and Designs	56
5.1 GasP NOR Gate t_{plh} vs NI in Different Technologies	56
5.2 Other Self-Timed Designs	57
5.3 Additional K Nodes in GasP	64
Chapter 6 Summary and Future Work	66
6.1 Randomness and Unique Signatures	66
6.2 Data Encryption	73
References	79

List of Figures

Figure 1: Drafting analogy.....	3
Figure 2 Clustering of tokens in a circular FIFO observed on one state-wire.....	3
Figure 3: Basic GasP circuit.	5
Figure 4: Two 5 gd loops in GasP.	5
Figure 5: Two configurations for the NOR in GasP.	7
Figure 6: A schematic timing diagram for two tokens passing through one GasP stage....	9
Figure 7: Three tokens and their intervals seen on a state-wire.....	10
Figure 8: Example interval fraction diagrams in a ring FIFO.	11
Figure 9: Ternary plot of three tokens from Figure 8 shown as trajectories.....	12
Figure 10: State-wire explanation for drafting.....	14
Figure 11: Drafting in an inverter chain.	15
Figure 12: A "crummy buffer" to offset drafting.....	16
Figure 13: The analog C-element after Fairbanks [17].....	17
Figure 14: The test circuit for investigating drafting.	19
Figure 15: Rail Pred vs. Rail Succ NOR configurations.	21
Figure 16: The NOR gate between tokens.....	22
Figure 17: Ideal transistors created in SPICE.	23
Figure 18: Ideal INV and NOR gates from ideal transistors.	23
Figure 19: Baseline drafting FIFO (above) and SW driver substitution (below).	24
Figure 20: All elements substituted and there is no drafting (top).	25
Figure 21: NOR test bench used for the Rail Pred NOR simulation measurements.	26
Figure 22: Simulation measurements of NOR in a test bench.....	27
Figure 23: NOR t_{plh} and V_K as a function of input interval (NI).	29
Figure 24: The NOR circuit mechanism of V_K decay.	31
Figure 25: The GasP NOR configuration determines drafting behavior	32
Figure 26: NOR propagation delay t_{plh} vs NOR input arrival time difference	35
Figure 27: SPICE and drafting model predictions.....	37

Figure 28: Fitting the simplified model for V_K to data from HSPICE simulation.	38
Figure 29: PMOS test Bench.	39
Figure 30: PMOS current (I_D) vs gate voltage (V_g) for different body voltages (V_b).	40
Figure 31: Threshold voltage (V_T) and V_K	40
Figure 32: The MATLAB event-driven model of GasP.	42
Figure 33: The difference equations for three drafting tokens in a ring FIFO.	43
Figure 34: SPICE and MATLAB simulations on a ternary graph.	44
Figure 35: Technique for injecting 3 tokens at precise intervals.	46
Figure 36: Interval algebra to find the line of demarcation.	47
Figure 37: Line of demarcation demonstrated with multiple MATLAB simulations.	49
Figure 38: One example of longer first interval collapsing first on a ternary graph.	50
Figure 39: The intended effect of the control circuit.	52
Figure 40: Details of the control circuit for node K.	53
Figure 41: Node K waveforms generated by the control circuit.	54
Figure 42: Control circuit results.	55
Figure 43: Comparing t_{plh} vs NOR Interval (NI) in three technologies.	56
Figure 44: Four gate designs, with inputs A and B.	58
Figure 45: Propagation Delay (t_p) vs Input Interval for different decision gates.	59
Figure 46: CEL operation in Micropipeline.	60
Figure 47: One Click stage from a test FIFO.	61
Figure 48: One stage of Mousetrap FIFO.	61
Figure 49: Phases of XOR operation in the Mousetrap implementation.	62
Figure 50: Two additional K nodes in GasP when using GOTW keepers.	64
Figure 51: Get-out-of-the-way keeper Pred driver K node.	65
Figure 52: Very slow drafting due to get-out-of-the-way keepers in GasP.	65
Figure 53: The shuffle circuit.	67
Figure 54: The drafting detector circuit.	69
Figure 55: The GasP demand merge stage.	70
Figure 56: Shuffle circuit token stream.	71

Figure 57: Six Monte Carlo simulations of the shuffle circuit.	72
Figure 58: Concept of encrypting interval data by drafting.....	73
Figure 59: Inverting the drafting curve to simplify the diagram.	74
Figure 60: An inverted anti-drafting curve will not perfectly reverse drafting.	75
Figure 61: Construction of a perfect anti-drafting curve.	75
Figure 62: Drafting two intervals.....	76
Figure 63: Two succeeding intervals undergoing anti-drafting.....	77
Figure 64: Results after attempted reversal of drafting for two intervals.	77

Glossary

Asynchronous and *Self-timed* are used interchangeably. This refers to circuits that clock themselves rather than relying on a global clock.

Circular FIFO (ring FIFO) - A linear FIFO with the ends connected. Because of the speed of these circuits and that some of the drafting effects occur over many stages, recirculating tokens in a ring makes the circuit more manageable. Circular FIFOs also have an additional interesting behavior. The sum of token spacings around the ring is constrained. This constraint forces tokens to interact in a way that can be displayed in a useful, graphical manner.

Empty state-wire - A state-wire is discharged to zero. Empty means no token present

Event - Passage of a token from one stage to another in a FIFO.

FIFO - First in first out buffer. For self-timed circuits, each stage of the FIFO operates independently, depending only on presence or absence of immediately upstream or downstream data.

Full state-wire - A state-wire is charged to V_{DD} . Full means a token is present.

GasP - A family of self-timed circuits so named by Sutherland and Fairbanks [1] which is based on Molner's "Asynchronous Symmetric Persistent Pulse Protocol, asP*" design [2].

Hand-shake signal and *Token* are used interchangeably. The non-data component of the FIFO event indicates the presence of data and is used in the transfer of data from one stage of the FIFO to another.

Mostly-empty FIFO - Less than maximum throughput FIFO occupancy. This is a function of forward and reverse latencies of the individual stages in the FIFO. With 6/4 GasP this is 60% of full.

Mostly-Full FIFO - More than maximum throughput FIFO occupancy.

Predecessor (Pred) - The state-wire connected to the input of a stage. Tokens arrive at a stage on that stage's predecessor.

Rail Pred - The option of connecting the inputs to the NOR gate in GasP where the rail-connected PMOS is controlled by the Predecessor.

Rail Succ - The option of connecting the inputs to the NOR gate in GasP where the rail-connected PMOS is controlled by the Successor.

Self-timed (ST) and *Asynchronous* are used interchangeably. This refers to circuits that clock themselves rather than relying on a global clock.

Stage - One basic self-timed unit in a FIFO, not including data controlled by the unit. Each stage operates autonomously following the logic of the handshake protocol. In this dissertation a stage passes no data, only a token.

State-Wire (SW) - Predecessor or Successor wires that connect one stage to another. The state-wires store the tokens as a charge to V_{DD} . Keepers are used to maintain the token indefinitely on a state-wire until the next stage can use it. The preceding stage places the token on the state-wire and the succeeding stage will remove the token and pass it forward along the FIFO.

Successor (Succ) - The state-wire connected to the output of a stage. Tokens depart a stage on its successor. One stage's successor is the next stage's predecessor. Each stage moves a token from its predecessor to its successor.

Token and *Handshake-signal* - are used interchangeably. The non-data component of the FIFO event. This indicates the presence of data and is used in the transfer of data from one stage of the FIFO to another.

t_{plh} - The low-to-high propagation delay through the NOR gate in GasP. The controlling input is the last logic low input and the output is then forced logic high.

Chapter 1 Introduction and Background

Self-timed circuits move data through pipelines using handshake signals rather than a global clock. Data and the handshake usually move together as bundled data. In this investigation, only the handshake signals are of concern. The sequence of handshakes along a chain of stages, or “FIFO”, may be regarded as the movement of “tokens” along the chain. The presence of a token between stages is indicated by a logic high on the state-wire connecting them. A token is absent if the state-wire is logic low. A token will advance through a stage if its input or “predecessor” state-wire is high, indicating presence of a token, and its output or “successor” state-wire is low, indicating a space or “vacancy”. This condition produces a pulse on the fire signal that advances the token [3, 4]. Tokens can change their spacing on the FIFO but not their sequence. If the FIFO is closed into a ring, tokens can recirculate continuously for testing purposes [2]. Except for experiments like ours, FIFOs are rarely closed into rings.

Because token movement is controlled by local handshake signals rather than a global clock, intervals between tokens can vary. When following tokens tend to catch up with leading tokens (Figure 1) the effect is called “drafting” after the technique bicyclists use to make cross-country cycling easier. The lead rider blocks the wind making it easier for the followers. In the FIFO the lead token causes changes in the circuitry which makes the trailing tokens propagate faster, eventually catching up to the lead token. Ultimately, they form a minimally spaced pair that moves as a group. If there are several tokens they will gather into a cluster with minimal spacing between tokens and that cluster will move as a unit (Figure 2). The opposite effect, where tokens are pushed apart, is called

“negative drafting” or “anti-drafting.” For drafting (anti-drafting) a lead token creates a memory condition in each stage where a following token propagates through the stage faster (slower). Total drafting is an accumulation of many small decrements in token intervals by each stage, so the amount of drafting by a following token depends on the number of stages traversed by the token as well as proximity of the preceding token. The drafting phenomenon is well-known and easily observed in circuit simulations such as SPICE and in silicon. So it is not necessary to invoke device physics beyond effects already embodied in SPICE models to observe the phenomenon and elucidate the circuit mechanism. In spite of this, there is controversy about exactly where the memory condition resides and how it works at the circuit level.

For many applications of self-timed circuits, drafting is unimportant because only the sequence of tokens matters and not the intervals between them. But there are new applications, such as spiking neural networks [5] or time-of-arrival measurements [6], for which the intervals carry meaning. The use of self-timed circuits in these applications will require control of drafting. The drafting effect investigated here is absent from synchronous circuits precisely because a global clock controls the arrival times of data tokens.

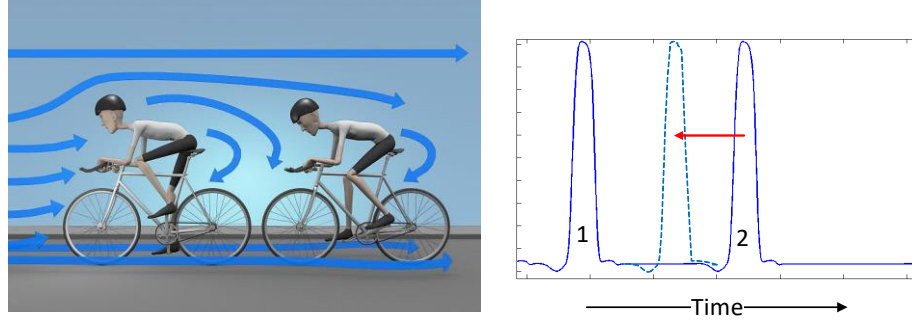


Figure 1: Drafting analogy. (left) Drafting in bicyclists. (right) Drafting in a FIFO observed on one state-wire.

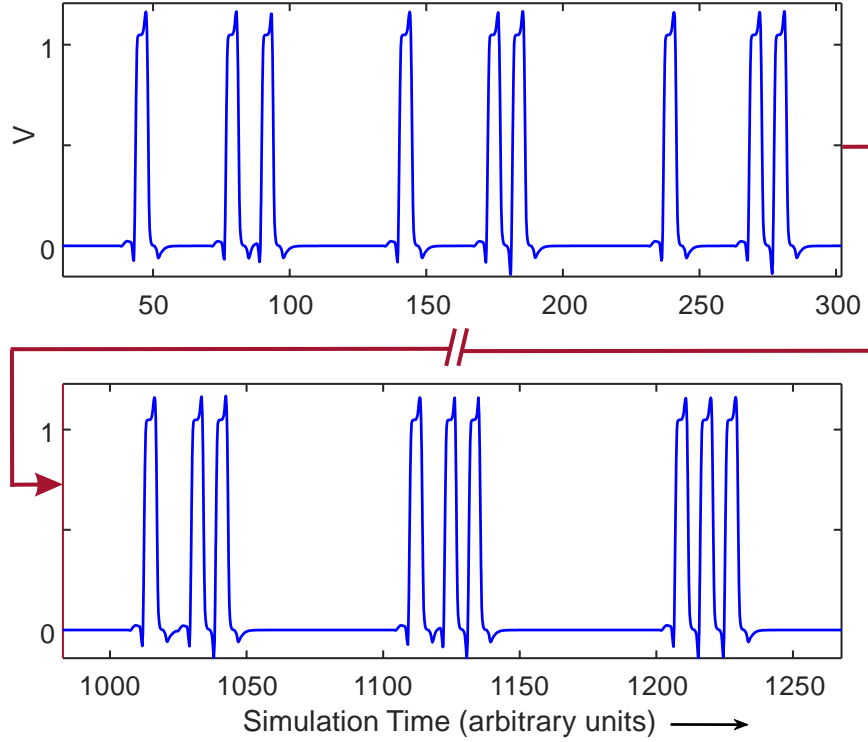


Figure 2 Clustering of tokens in a circular FIFO observed on one state-wire.

1.1 Basic Gasp Operation

Experiments in drafting were performed using GasP [1] because of our familiarity and experience with this stage design. Tokens arrive on the predecessor (Pred) state-wire and

exit on the successor (Succ) state-wire. The decision gate has a logical AND function which detects an empty Succ and a full Pred which is the condition for GasP action. In GasP the AND function is implemented as the De Morgan's equivalent NOR gate. The inputs to the NOR are $\overline{\text{Pred}}$ and Succ signals. When both inputs are logic LOW then the NOR generates an output HIGH and a Fire signal is initiated.

The design uses 6/4 GasP which has a forward delay of 6 gate delays (gd) and a reverse of 4 gd (Figure 3). This means that if a token appears on the Pred and there is no token on the Succ the circuit will “Fire” and move the token to the successor in 6 gate delays. If there is a token on both the Succ and Pred and the Succ token is then consumed by the next stage, a “vacancy” is created on the Succ. That vacancy will allow the token on Pred to move forward to the Succ. The vacancy therefore moves backwards in 4 gate delays which is faster than a token moving forward. This ratio of forward to reverse delay has been shown to improve throughput in GasP FIFOs because it takes longer to move bundled data forward to the next stage than it is to signal “no data” back to the previous stage. Therefore, it makes sense to make the backwards latency shorter.

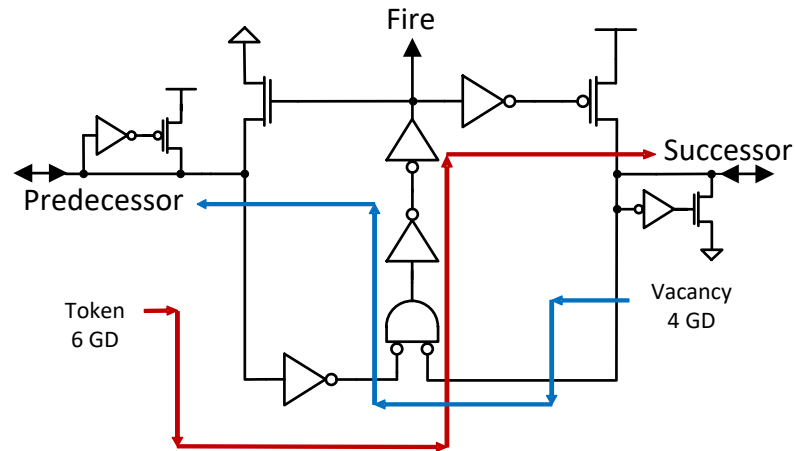


Figure 3: Basic GasP circuit. A token will propagate forward (through) the stage in 6 gd. A vacancy propagates backwards in 4 gd.

There are two 5 gd loops in GasP (Figure 4). The duration of the loops ensures reliable operation and full charging and discharging of the state-wires. The Fire pulse duration is 5 gd. If there are tokens on both Pred and Succ state-wires, the removal of the Succ token activates the Succ-side (red) loop moving the Pred token to the now empty Succ. If Pred and Succ are empty and a token arrives on the Pred the Pred-side (blue) loop will be activated moving the Pred token to the Succ.

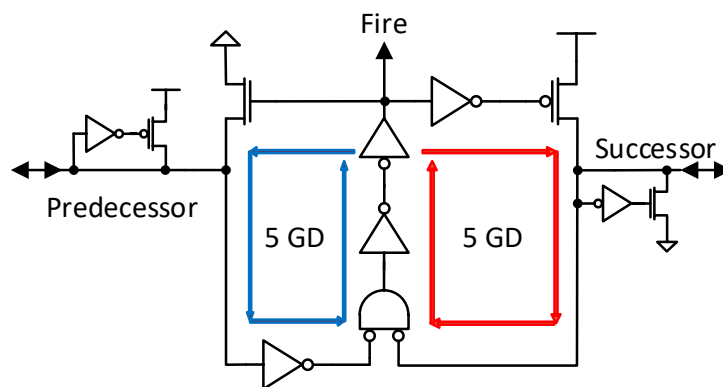


Figure 4: Two 5 gd loops in GasP. The 5 gate delays ensure that the state-wire drivers will fully charge and discharge the state-wires.

Figure 5 shows two NOR configurations that can implement the logical AND logic function: “Rail Pred” and “Rail Succ”. In the Rail Pred configuration, the NOR input that controls the rail-connected PMOS transistor is connected to the predecessor side of the GasP unit. The other NOR input is then connected to the successor side which controls the output connected PMOS. The Rail Succ configuration has the inputs reversed so that the rail-connected PMOS transistor is connected to the successor side.

The “K”-node between the PMOS transistors is key to drafting. The slow decay of the charge on K between tokens changes the propagation delay (t_{plh}) through the NOR which modifies the interval as the tokens pass through the GasP stage.

The event of a token passing through a GasP stage is marked by the Fire signal which is derived from the NOR output. Fire is typically used to strobe data through latches, but in these experiments, latches are omitted, and the focus is only on the handshake events. Many GasP circuits use “get-out-of-the-way” (GOTW) keepers that are disabled when the state-wires are changed. But the experiments reported here use always-on over-driven keepers for simplicity. This GasP design is therefore simple but shows all the drafting behavior of interest. GOTW keepers will be discussed later.

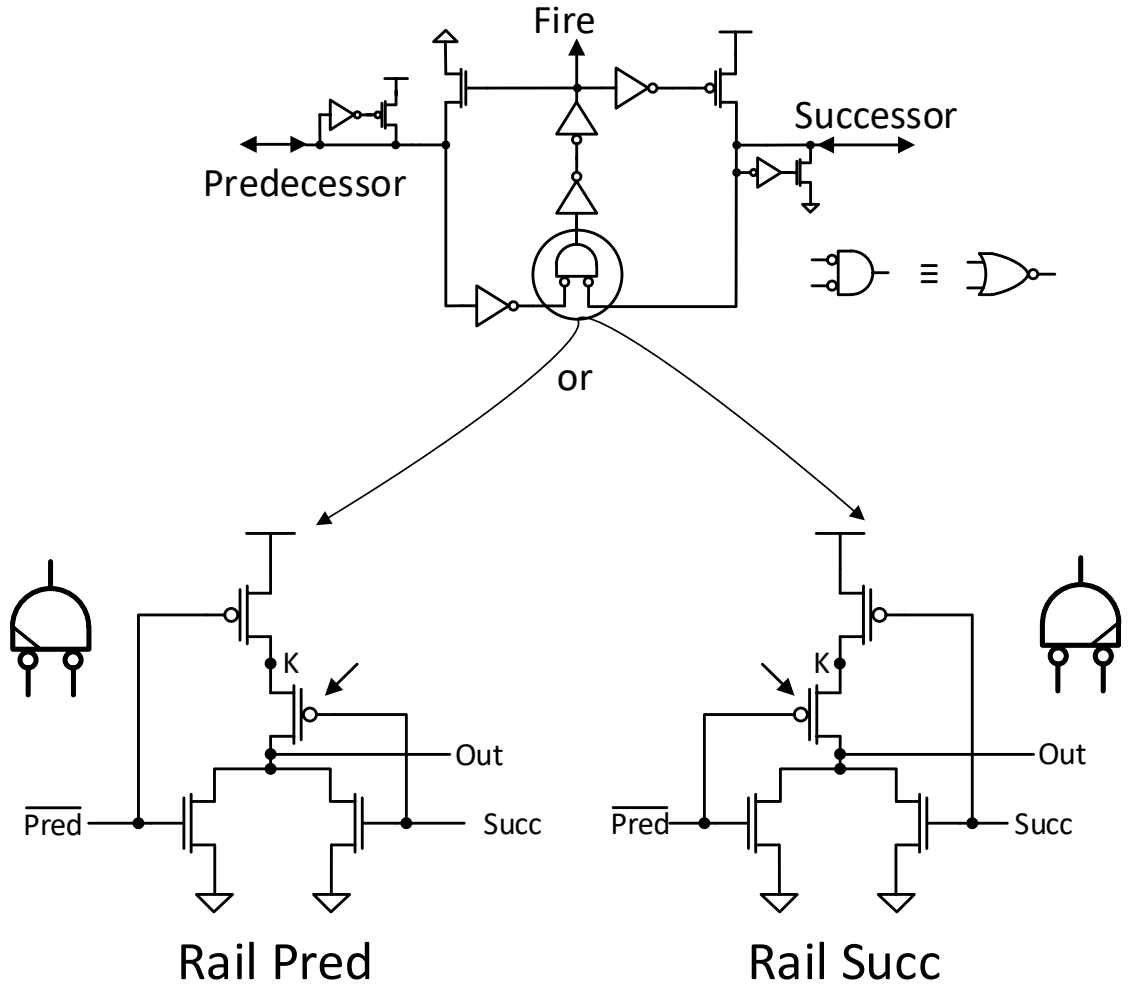


Figure 5: Two configurations for the NOR in GasP. Depending on which input controls the rail-connected PMOS the configurations are Rail Pred and Rail Succ. The diagonal bar in the gate icon indicates which input is rail-connected.

1.2 Basic GasP Timing

A schematic timing diagram, Figure 6, shows two cases of a pair of tokens passing through one Rail Pred configured GasP stage in an initially quiescent FIFO. In either case the tokens are taken to be separated more widely than minimum spacing so that they are not yet fully drafted. That is, $TI > 10$ nominal gate delays. The token interval, TI , is

the interval between tokens measured between rising edges of Pred . Also, the cycle time of 10 nominal gate delays is required for the logical processing to move a token across a GasP stage and reset the stage. The NOR interval, NI , is the interval between inputs to the NOR gate; that is, between the falling edge of Succ due to the leading token, and the falling edge of $\overline{\text{Pred}}$ due to the following token Eq. (1). NI is part of TI , and TI is 10 nominal gate delays longer than NI . One of the 10 nominal gate delays is the delay through the NOR gate, t_{plh} , timed from the controlling falling input of the NOR Eq. (2).

$$NI = t(\overline{\text{Pred}} \downarrow) - t(\text{Succ} \downarrow) \quad (1)$$

$$t_{plh} = t(\text{NORout} \uparrow) - \max[t(\overline{\text{Pred}} \downarrow), t(\text{Succ} \downarrow)] \quad (2)$$

During the NOR interval (NI), the K-node is isolated so the voltage V_K on that node decays slowly. Comparison of Figure 6(a) and Figure 6(b) shows that when NI is shorter, the decay of V_K is less complete so the subsequent NOR t_{plh} transition is shorter because it starts from a potential closer to V_{DD} . That is, for shorter NI , the following token is passed more quickly through the GasP stage than for longer NI . In the two-token example the leading token transits the GasP with maximum delay corresponding to $t_{plh}(NI = \infty)$, so the following token always transits the GasP more quickly than the leading token. The following token therefore catches up with the leading token (reducing TI and NI), with the rate of catch-up increasing as the following token closes in on the leading token. Once $TI \approx 10$ nominal gate delays ($NI \approx 0$) the following token stops catching up with the leading token and moves at the same speed as the leading token.

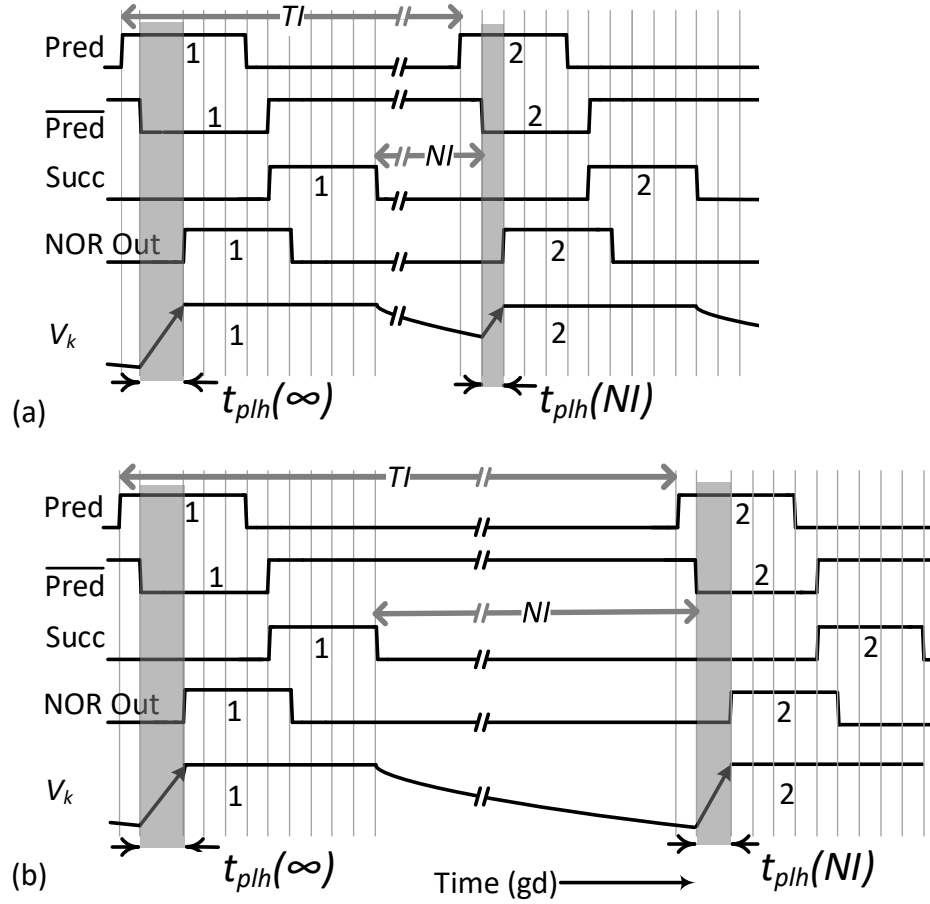


Figure 6: A schematic timing diagram for two tokens passing through one GasP stage. The stage uses a Rail Pred NOR. Token 2 follows token 1. (a) shows a short TI and a short NI resulting in a short NOR t_{plh} . Token 2 propagates through the GasP stage with minimal delay. (b) shows a longer TI and NI resulting in a longer t_{plh} and token 2 takes longer to propagate. K node voltage V_K decays between the fall of Succ and the fall of $\overline{\text{Pred}}$ and alters t_{plh} .

1.3 Depictions of Drafting Behavior

There are several ways to depict drafting behavior. The simplest, Figure 7, shows three tokens arriving at the Pred state-wire of one GasP stage in a circular, mostly-empty FIFO. Tokens are observed as they pass on this state-wire. The intervals are the time difference between token arrivals. Interval A is the time between token 1 and token 2.

Interval B is between token 2 and token 3. Interval C is between token 3 and the recirculating token 1.

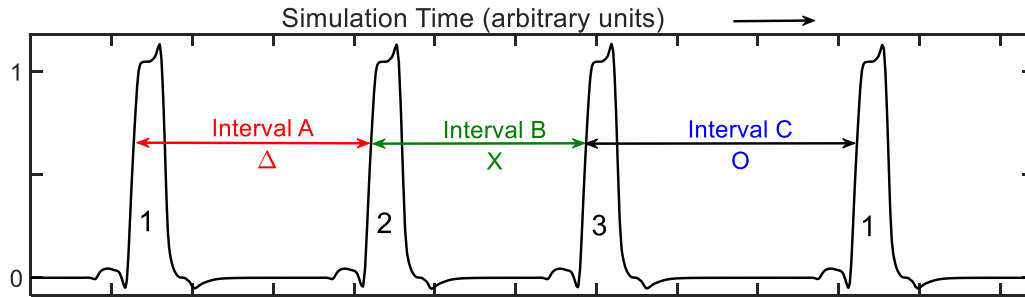


Figure 7: Three tokens and their intervals seen on a state-wire.

Since it is the token intervals (TIs), rather than the tokens themselves, that are important in drafting, plotting TIs directly shows drafting more clearly. The TIs A, B, and C can be plotted separately to show the evolution of each TI over time. In a ring FIFO, each interval can also be expressed as an interval fraction (IF) of the total interval sum thereby normalizing the intervals to a fraction of unity. In the drafting example, Figure 8(left), tokens start at arbitrary IFs of $A = 0.29$, $B = 0.28$ and $C = 0.52$. Over time C grows to a maximum as A and B shrink to a minimum. When fully drafted A and B = 0.1 which is the minimal cycle time through one FIFO stage. This is set by the logic delays of the circuit. IF C is then 0.8. In the anti-drafting example [Figure 8(right)] a hypothetical control circuit is used to spread the tokens apart rather than draw them together into a cluster. The three tokens start fully drafted from Figure 8(left) and with time the IFs become equal. This equal-spacing is the result of full anti-drafting. One can also plot the

IFs from a circular FIFO on a ternary graph (Figure 9) because the three interval fractions must always sum to one. The interval trajectories are taken from Figure 8.

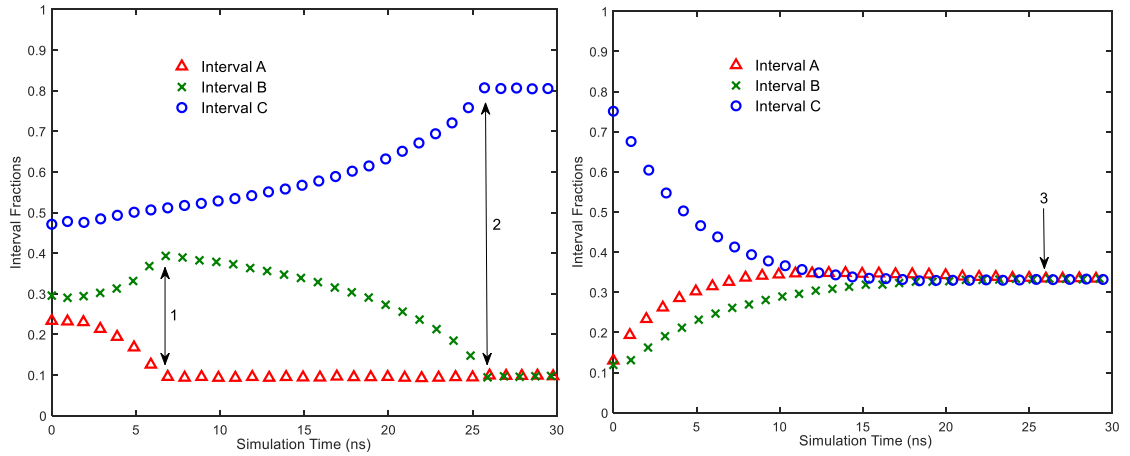


Figure 8: Example interval fraction diagrams in a ring FIFO. Drafting (left) shows collapse of IF A and B to minimum and IF C increasing to maximum. Anti-drafting (right) is a continuation of the end of the drafting diagram (left) and shows all IFs becoming equal. Points 1-3 also appear in Figure 9.

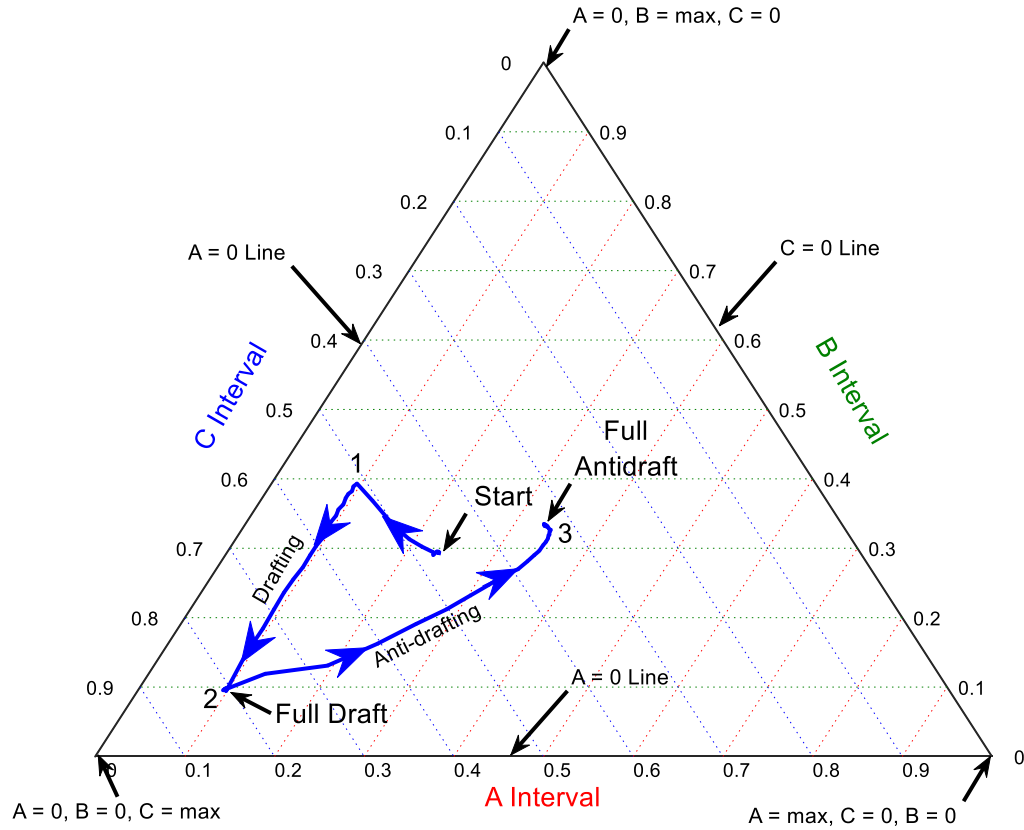


Figure 9: Ternary plot of three tokens from Figure 8 shown as trajectories. From the arbitrary starting point, first the A IF collapses to a minimum of 0.1, then the B IF collapses to the same minimum. IF C is then at a maximum of 0.8. IFs must sum to 1. Points 1-3 are the same as in Figure 8.

1.4 The Conventional Explanation

Two explanations for token interaction in a self-timed FIFO have been given [7, 8]. The first is the added delay in the two-input decision gate when the inputs are nearly simultaneous [9], also called the “Charlie” effect in [10]. The Charlie effect is significant only when tokens are nearly minimally spaced which occurs near maximum FIFO throughput (see Figure 23). The minimum spacing of tokens in a GasP FIFO is the result of the cycle time of GasP and only is slightly modified by the Charlie effect.

Applications that use intervals between tokens for data cannot operate with minimum token spacing; the data would be lost because the intervals would be replaced by a fixed circuit-dependent interval. It is remarkable that drafting occurs even for widely spaced tokens well beyond the range of the Charlie effect. Although valid and observable the Charlie effect has no significant role in long-range drafting.

The other explanation for token interaction is the drafting effect which is attributed to state-wire capacitance [11]. This “state-wire drafting” is explained as the result of incomplete state-wire charging or discharging which speeds up transitions when tokens are more closely spaced (Figure 10). [8]. During the brief firing event in GasP the state-wire driver incompletely charges or discharges a state-wire. After the Fire smaller keepers finish the charging or discharging to the power rails. The combination of state-wire drafting and the Charlie effect is frequently used to describe token interaction in self-timed ring oscillators [12-17].

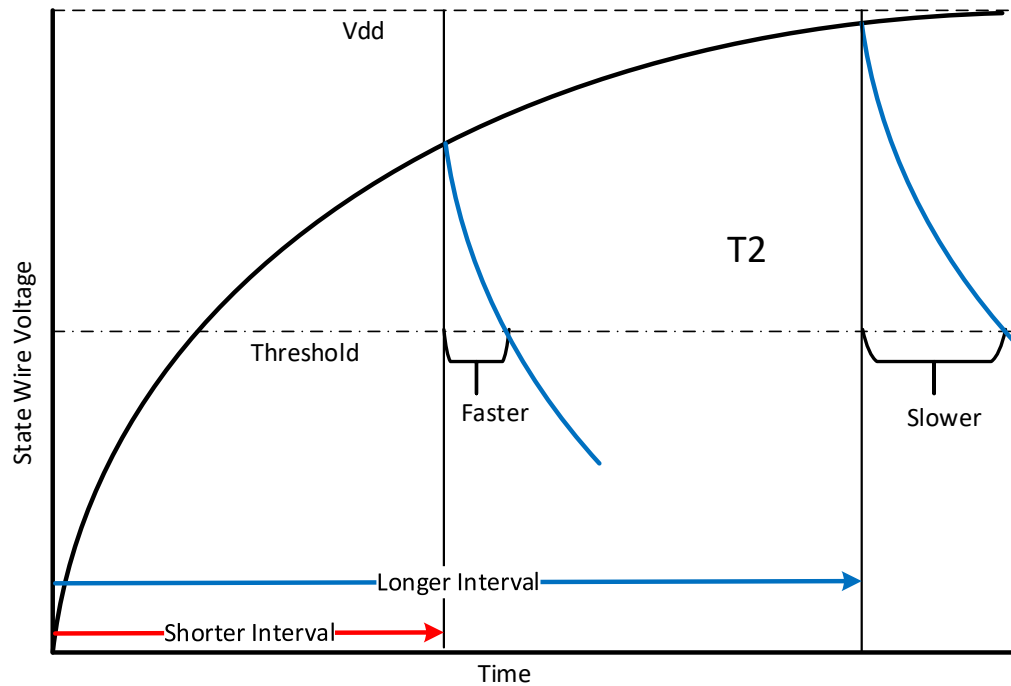


Figure 10: State-wire explanation for drafting. State-wires are incompletely charged or discharged before the next state-wire change. For shorter intervals the change can be made faster which speeds up propagation. For longer intervals the change is slower which slows propagation.

The state-wire drafting explanation, incomplete charging and discharging of the state-wires, does not account for the drafting effect observed for inter-token intervals of several thousand gate delays. It is implausible that state-wires would be incompletely charged or discharged over such long intervals. State-wire drivers in GasP are driven for 5 gate delays which is more than enough time for a properly designed state-wire driver to fully charge or discharge any realistic state-wire capacitance. Additionally, SPICE simulation shows no measurable drift in state-wire potential after the drivers are turned off.

Yet another drafting-like phenomenon occurs when a chain of inverters is operated near their maximum operating frequency. Figure 11 shows that if a pulse stream is presented

to the inverter chain the first interval, which is between the first pulse and the second pulse, will shorten as it passes through the inverter chain if the inverter chain is driven near its maximum operating frequency. This only occurs for the first interval and only at the edge of reliable operation of the logic. This is due to the incomplete charging and discharging of the interconnect between inverters. Note that in GasP no gate or transistor is operating at its maximum frequency. So, this inverter-chain-like drafting does not need to be considered in normal GasP operation.

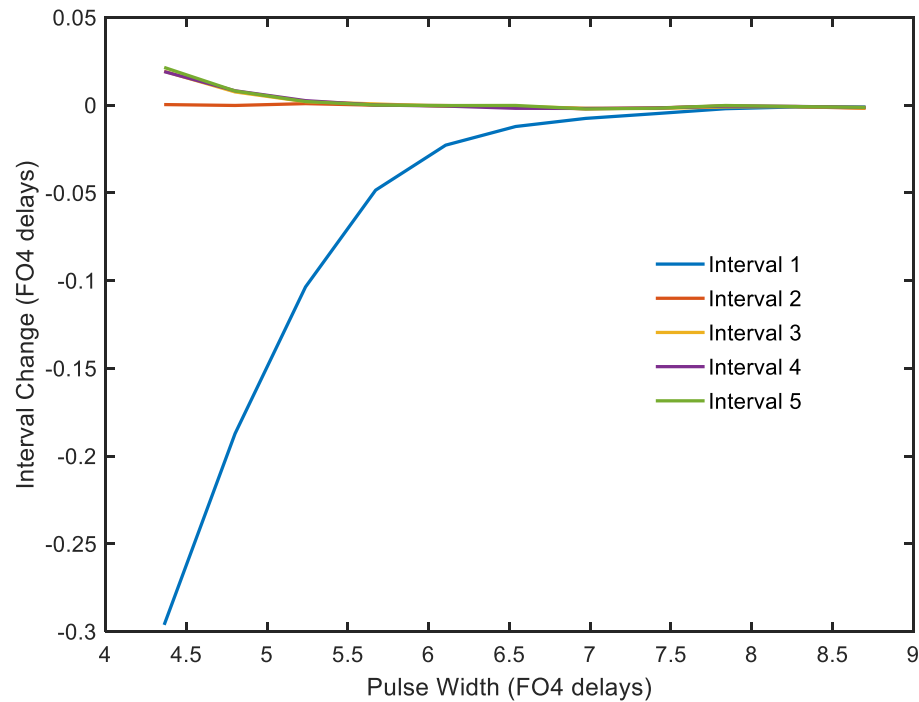


Figure 11: Drafting in an inverter chain. The first interval (between pulse 1 and pulse 2) can shorten if the pulse width is narrowed to the edge of operation.

1.5 Conventional Mitigation Approaches

The mitigation strategies offered in the literature mitigated drafting by introducing a reverse delay profile through the decision gate. This cancelled the variable delay in FIFO stage propagation that is responsible for drafting. One approach [8] to reverse or stop drafting is to introduce a “crummy buffer” after the C-element in a Micropipeline FIFO. The crummy buffer creates the inverse profile for the C-element in Micropipeline (Figure 12). The reverse sloping profile can cancel the opposite state-wire slope thereby nullifying drafting. If the reverse slope is greater than the conventional slope, anti-drafting is achieved. The authors even achieved anti-drafting using the crummy buffer, but this required fine tuning of I_{ref} to achieve the precise feedback slope.

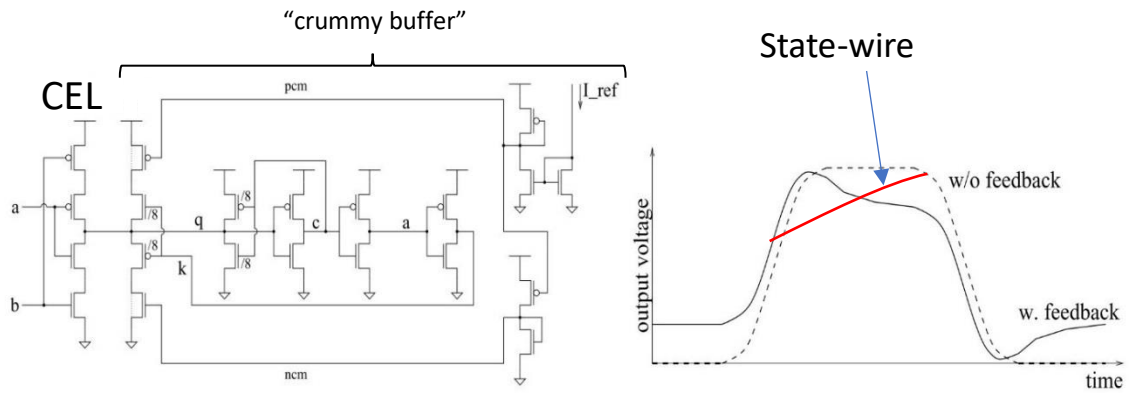


Figure 12: A "crummy buffer" to offset drafting (from [8]). The engineered and finely tuned waveform compensates for the purported waveform on the state-wire thereby eliminating and/or reversing drafting. Another approach to mitigate drafting [17] was to create an “analog CEL”

for Micropipeline which does not draft. The decision gate (CEL) is replaced by an inverter combination (Figure 13) which results , essentially, in a ratioed gate. This

nullification of drafting allowed the author to then more easily achieve equally spaced tokens for timing purposes.

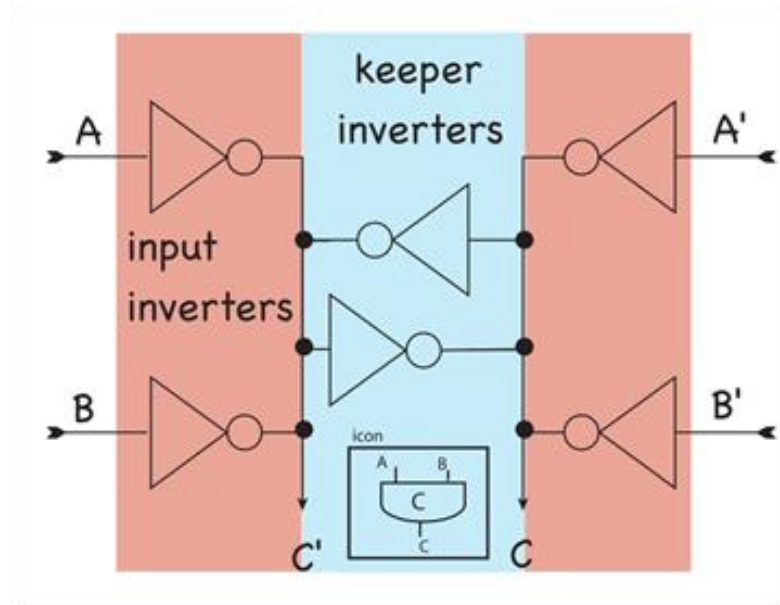


Figure 13: The analog C-element after Fairbanks [17].

A key feature of this circuit is that it avoids internal nodes in the CEL decision gate which the author asserts can “cause unexpected phase shifts.”

Both approaches were successful in mitigating drafting but did not address the cause directly, which is the behavior of the internal node in the decision gate, because they were focused on the state-wire capacitance as the cause.

Chapter 2 Strategy

Noting that the conventional explanation for drafting is suspect, several experiments were designed in SPICE to find the real cause. The first is a test FIFO to observe drafting and probe the GasP stages. The second is a substitution study where successive transistors in GasP stages are replaced with ideal switches to find the location for the cause. The site for the cause of drafting is in the GasP NOR gate. The third is a NOR test-bench to characterize NOR behavior in more detail.

2.1 The Basic GasP FIFO

A first experiment explores the behavior of the 17 stage GasP FIFO shown in Figure 14. A prime number of stages was chosen to avoid any synchronization artifacts. All Rail Pred and all Rail Succ NOR configurations were separately simulated. The generic 32 nm Synopsys model in HSPICE was used for simulation at nominal temperature (25° C) and supply voltage (1.0 V). Logical effort was used to size the gates in GasP so that each gate has the same delay and near-optimal circuit performance. Timing is reported in gate delays (gd) which is the nominal propagation through an FO4 inverter; in this technology $1 \text{ gd} \approx 10 \text{ ps}$.

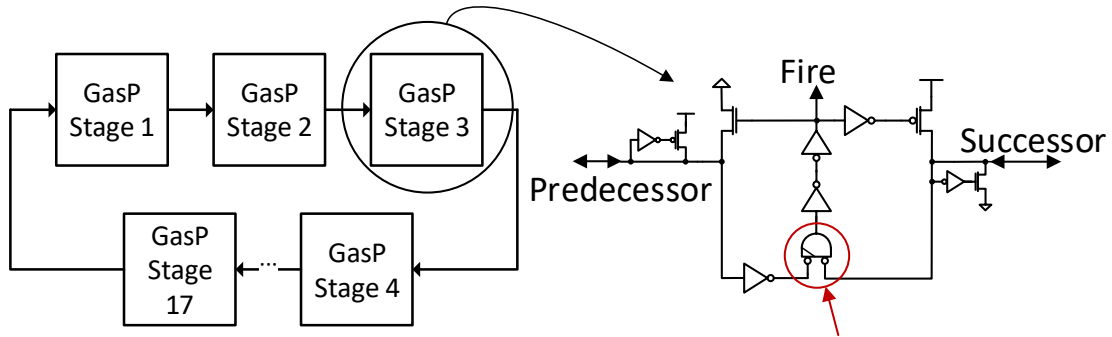


Figure 14: The test circuit for investigating drafting. The Rail Pred NOR configuration (circled) is shown here.

In a “mostly empty” FIFO the tokens are free to move and are not fully drafted. On the other hand, a “mostly full” FIFO contains many tokens minimally spaced with vacancies which are free to move and are not fully drafted. Vacancies draft like tokens but appear to move in the reverse direction of drafting tokens. Once fully drafted, tokens maintain a minimal spacing which is the cycle time of one GasP stage. Fully drafted vacancies coalesce into one larger gap while the tokens remain minimally spaced.

2.2 Initial Observations

A GasP FIFO containing sparse tokens was observed to draft by the mechanism shown in Figure 6 for the GasP NORs in the Rail Pred configuration, but the same FIFO using the Rail Succ NOR configuration would not draft. Figure 15 shows the behavior of the internal K node in the Rail Pred configuration (top) and in the Rail Succ configuration (bottom). The difference between Rail Succ and Rail Pred configurations is only in the behavior of the internal node of the NOR gate called “K.” Since K is not an output node it is not always actively driven and may drift between NOR actions. This changes the

delay through the NOR. All other nodes, including the state-wires, are actively driven and delay through all non-NOR gates is interval independent.

For Rail Pred, the voltage at node K, " V_K ", decays between tokens (Figure 6 and Figure 15, top), while for the Rail Succ V_K does not decay between tokens (Figure 15, bottom). The time difference between tokens determines when, during V_K decay, the NOR gate initiates the Fire signal. This will determine the propagation delay through the NOR, " t_{plh} ", and the delay through the GasP stage. If V_K does not decay between tokens, t_{plh} is constant because, unlike the Rail Pred case in Figure 15, t_{plh} does not depend on token spacing. The delay of a token propagating through a GasP stage depends on NOR t_{plh} which, in turn, depends on when the previous token exited the stage and how far V_K has decayed. Faster t_{plh} occurs when the previous token interval is short. When the trailing token moves faster the interval shortens further. Faster intervals shorten faster than longer intervals. This is the exact behavior in drafting and K node behavior offers a plausible explanation.

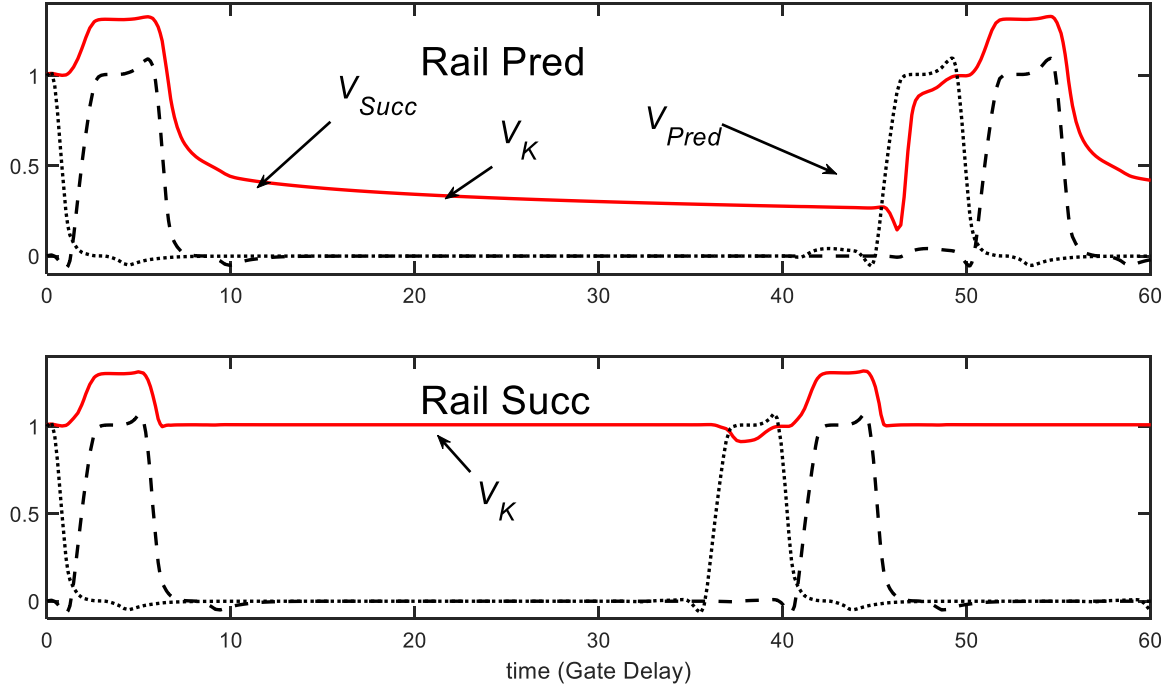


Figure 15: Rail Pred vs. Rail Succ NOR configurations. For Rail Pred, V_K decays between tokens, while for Rail Succ it does not. Decay of the K node residual charge affects NOR t_{pth} and ultimately delay through the GasP stage.

A closer look at NOR behavior between tokens (Figure 16) shows the cause for the decay of V_K . Between tokens Succ (input B) is logic LOW and \overline{Pred} (input A) is logic HIGH. In the Rail Pred configuration, V_K decays between tokens (Figure 16, left) because the lower PMOS transistor is diode connected and discharges K slowly through the \overline{Pred} - side NMOS transistor which is turned on. The lower PMOS transistor is diode-connected because both the drain (output) and the gate (input b) are at 0 V and are effectively connected. On the other hand, in the Rail Succ configuration, V_K is held constant at V_{DD} by the upper PMOS transistor which is turned on (Figure 16, right).

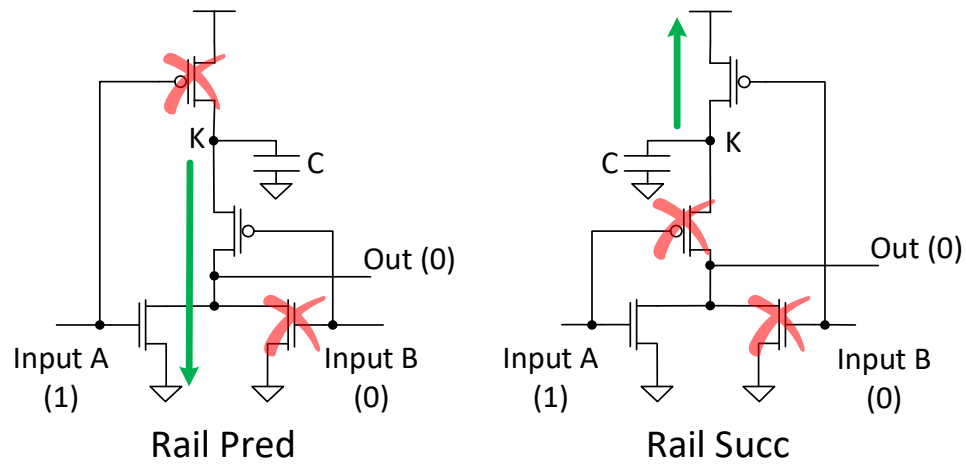


Figure 16: The NOR gate between tokens. With Rail Pred K discharges through a diode connected PMOS. With Rail Succ K is held constant at V_{DD} .

2.3 Substitution Study

The possible effect of state-wire capacitance on drafting was studied in a second experiment by substitution of ideal transistors for state-wire drivers in a simulation of GasP elements in a FIFO showing drafting. Ideal transistors eliminate state-wire capacitance charging delays by instantly fully charging or fully discharging the state-wires. Near ideal transistors were constructed from ideal delay and switch blocks (Figure 17). The switches have a 1Ω on resistance and a $1G\Omega$ off resistance. Since the switches operate instantaneously, a nominal 1 gd is added to mimic the delay in a real transistor.

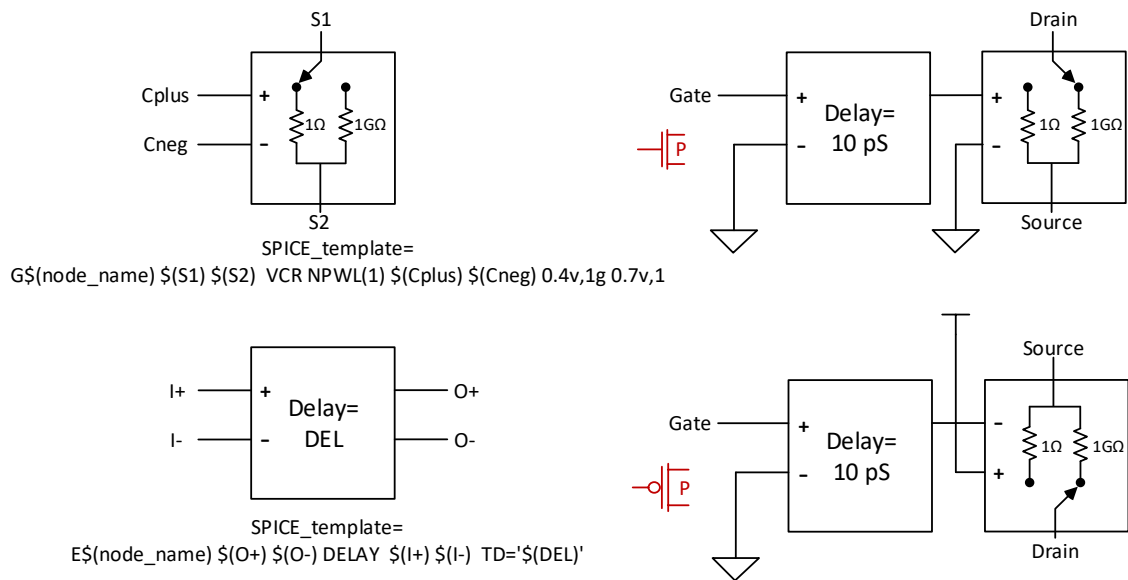


Figure 17: Ideal transistors created in SPICE. The “P” notation indicates “perfect.”

The result of the ideal transistor substitution for the state-wire drivers is shown in Figure 19. There was no change in drafting. This shows that the state-wires are not the cause of drafting. Note that a 1 pf capacitor was added to the state-wires. This is needed to eliminate erratic behavior of the ideal switches in SPICE.

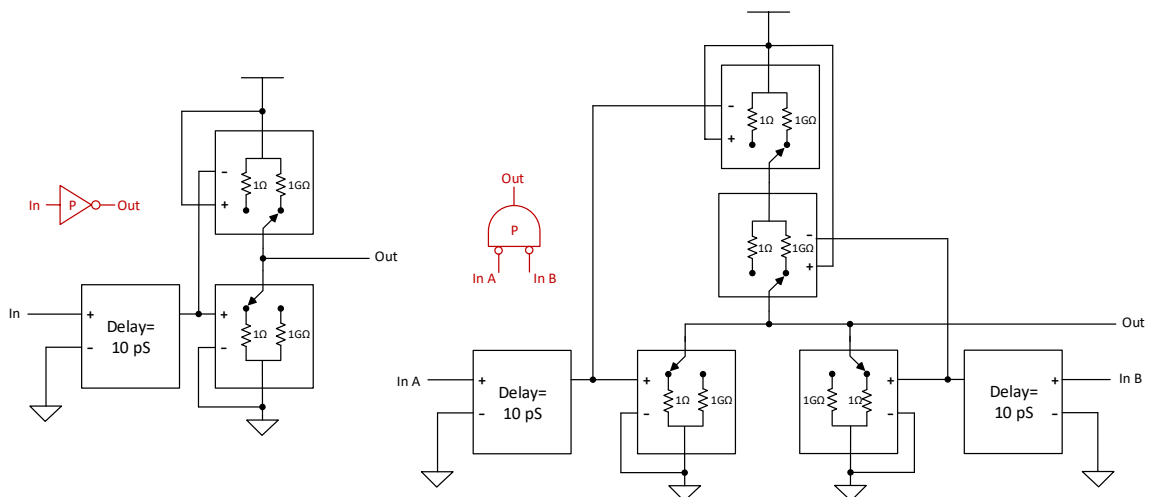


Figure 18: Ideal INV and NOR gates from ideal transistors.

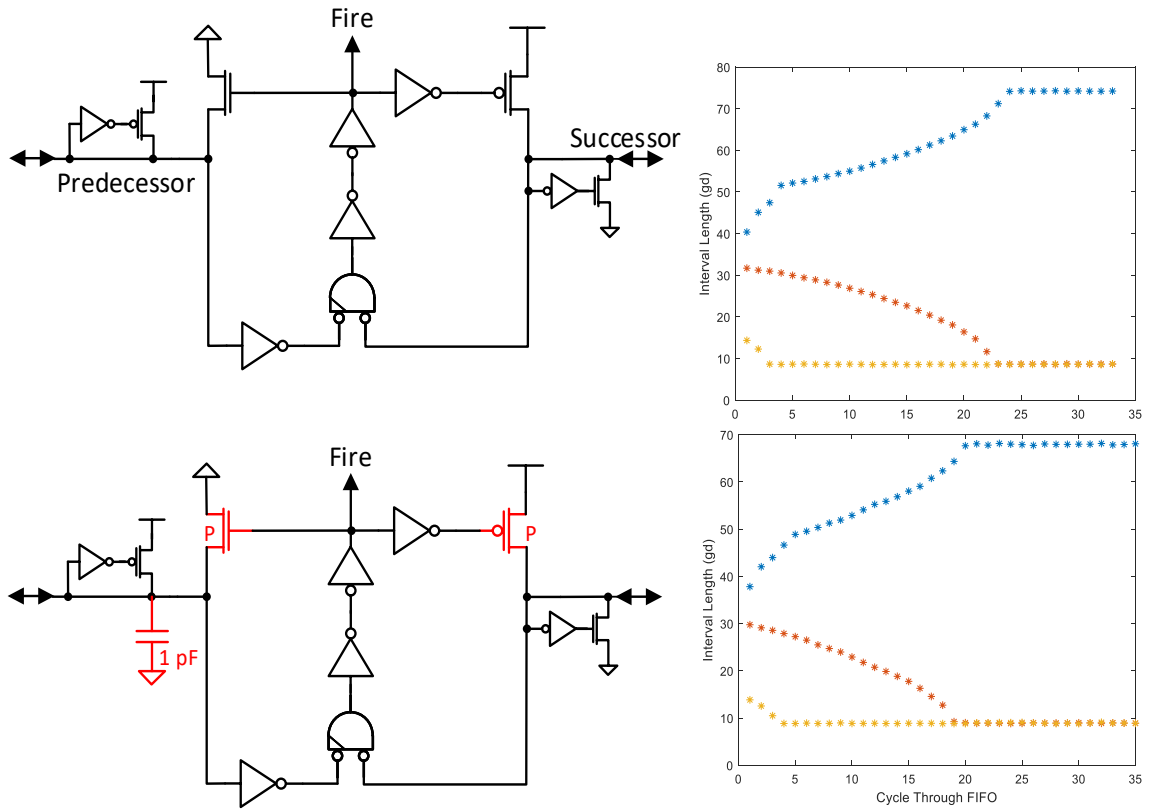


Figure 19: Baseline drafting FIFO (above) and SW driver substitution (below). There is no change in drafting.

The second experiment continued by constructing ideal inverters and an ideal NOR from ideal transistor building blocks (Figure 18). Continued substitution revealed the cause of drafting. Only when the ideal NOR was substituted for the “normal” NOR did drafting cease (Figure 20, top).

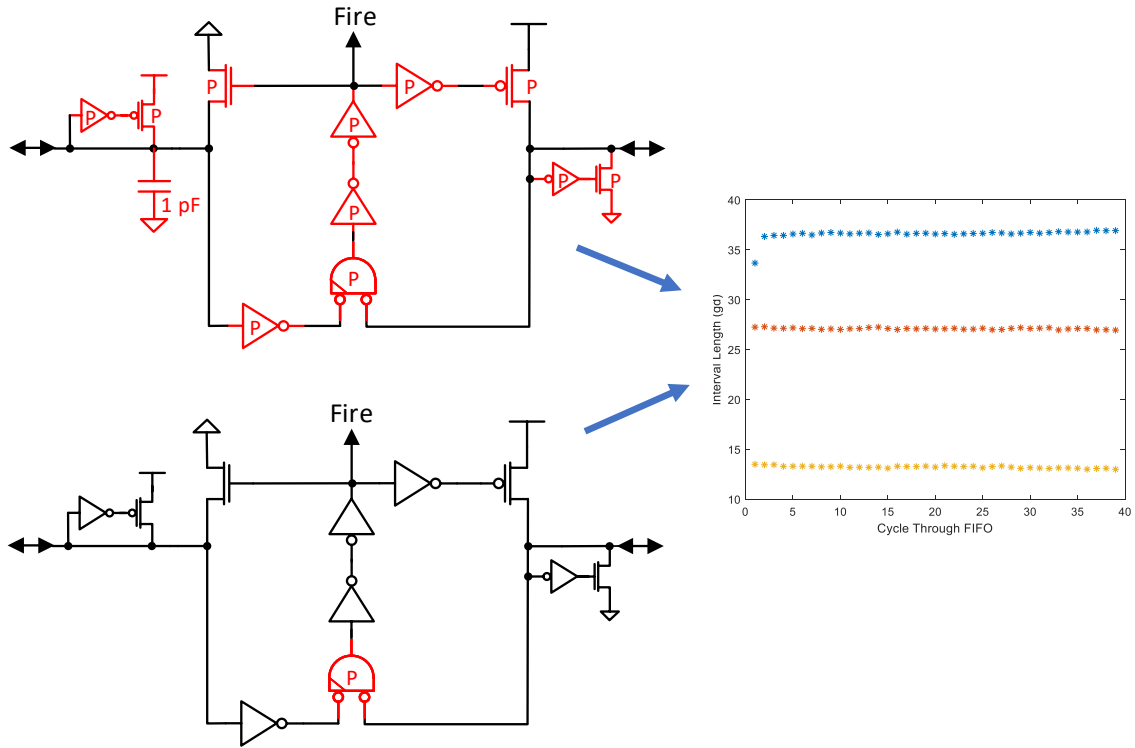


Figure 20: All elements substituted and there is no drafting (top). Just the NOR substituted and still no drafting (bottom).

Further substitution with ideal components within the NOR showed that drafting ceased only when either or both PMOS transistors in the NOR stack were replaced by ideal transistors (see Figure 16). This implicates the PMOS stack in the NOR and the behavior of the common K node as the key factor in the drafting mechanism. An ideal rail-connected PMOS transistor substitution disables drafting by pulling the K-node high quickly, regardless of the value of V_K . More fundamental to the elimination of the “memory-effect” of the drafting mechanism is the way an ideal output-connected PMOS transistor substitution disables the slow decay of V_K prior to when the $\overline{\text{Pred}}$ input goes low.

2.4 Measuring V_K and t_{plh} of the NOR Gate

In a third experiment a test bench was designed and used to measure the timing characteristics of an isolated Rail Pred NOR gate in a SPICE simulation. This offers a more detailed examination of the relationship between timing, V_K , and propagation delay plus the SPICE measurements are more accurate than using a running FIFO. The interval between NOR inputs (NI) was varied using an ideal SPICE delay element (Figure 21).

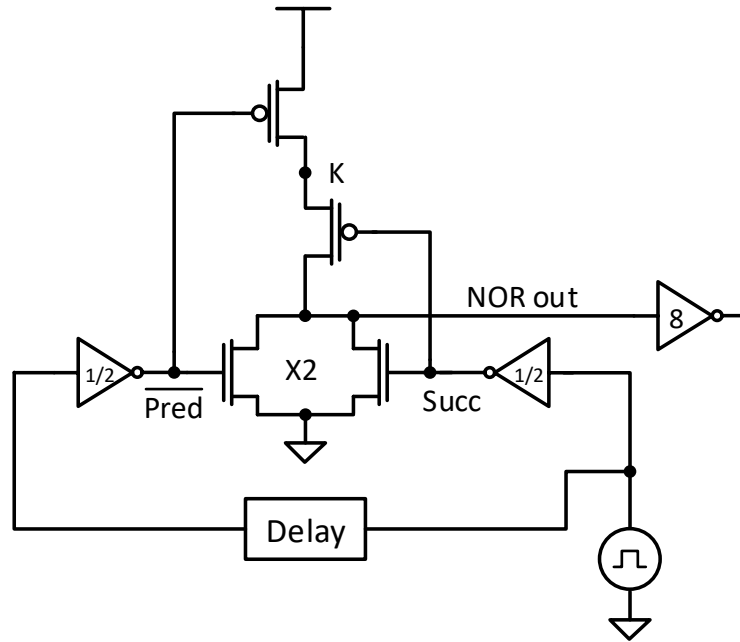


Figure 21: NOR test bench used for the Rail Pred NOR simulation measurements. A variable delay element is used. Sizing of components was chosen to give an FO4 loading milieu.

Both inputs to the NOR, the NOR output, and V_K were probed. NI was measured as the time difference between the 50% falling edges of $Succ$ and \overline{Pred} using Eq. (1). The NOR propagation delay (t_{plh}) was measured from the 50% falling edge of \overline{Pred} to the 50% rising edge of NOR output, which is the typical condition for a FIFO with sparse

tokens, using Eq. (2). The simulation measurements of V_K , NI , and t_{plh} for positive NI (when drafting is observed) are shown in Figure 22. Figure 22(a) shows that voltage V_K decreases as NI increases. Figure 22(b) shows that NOR t_{plh} increases as V_K decreases. Elimination of V_K between the top two graphs gives the bottom one which shows t_{plh} increasing with NI , Figure 22(c). This quantifies the effect shown schematically in Figure 6.

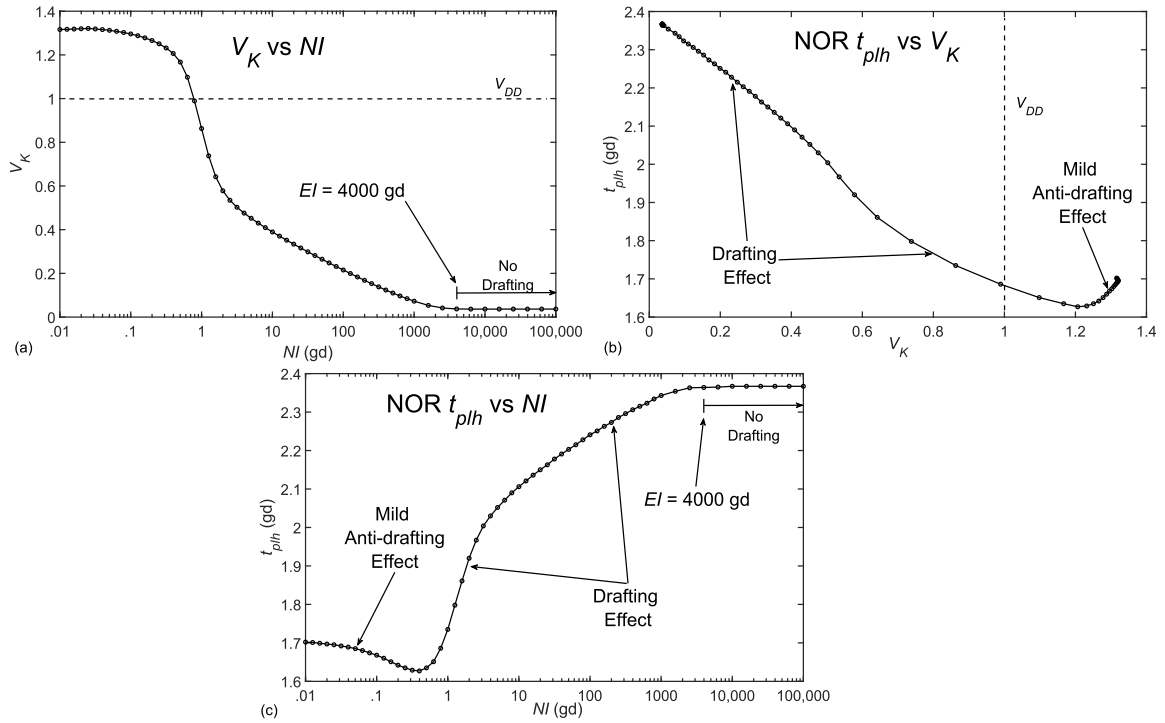


Figure 22: Simulation measurements of NOR in a test bench. (a) Voltage V_K decreases with NI . (b) t_{plh} decreases with V_K . (c) Elimination of V_K in (a) and (b) results in t_{plh} increasing with NI . Note the slight anti-drafting “Charlie” effect at very short NI (less than one gate delay).

NOR t_{plh} is approximately linearly related to V_K , Figure 22(b), because PMOS transistors approximate constant current sources when charging node K and the NOR output node.

The time to charge up to the threshold for generating a logic high at the NOR output will

also be linear with time. With short intervals ($NI < 1$ gd) node K remains at or above V_{DD} and t_{plh} is nearly constant. There is only slight anti-drafting due to the Charlie effect shown here. In this technology V_K is initially boosted above V_{DD} during NOR output high due to prominent Miller feed-through (Figure 15). The drafting effect extends to 4000 gd, the “escape interval” (EI). Beyond EI a trailing token is unaffected by the preceding token and there is no drafting. Since t_{plh} increases with interval this means that longer intervals do not shorten as much as shorter intervals which is the mechanism of drafting. These simulations show that drafting occurs even for token intervals up to 4000 gate delays (40 ns) which agrees with the duration of V_K decay.

NOR testbench measurements for both positive and negative NI are shown in Figure 23. For negative NI , t_{plh} is very different from positive NI . In a sparse vacancy FIFO $\overline{\text{Pred}}$ falls before Succ, so NI is negative, t_{plh} is independent of NI , and there is no drafting. On the other hand, in a sparse token FIFO, $\overline{\text{Pred}}$ falls after Succ like Figure 6, so NI is positive, t_{plh} increases strongly with NI , and drafting occurs.

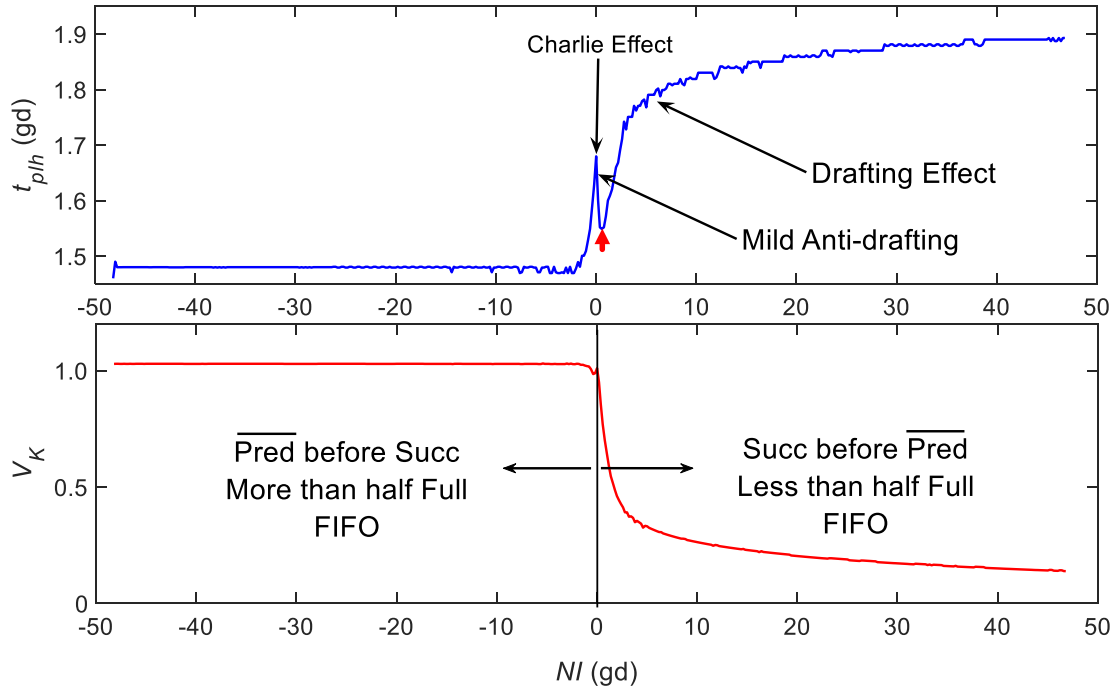


Figure 23: NOR t_{plh} and V_K as a function of input interval (NI). Red arrow indicates NI for maximum throughput spacing.

The shape of the t_{plh} curve for positive NI induces drafting because shortening the interval between tokens reduces the delay and shortens the interval further. Tokens separated by many hundreds of gate delays will draft by this mechanism. When inputs are nearly simultaneous a different behavior occurs. Near simultaneous inputs ($NI \approx 0$) there is a peak in t_{plh} which is the “Charlie” effect [10]. This is a well-known behavior where simultaneous inputs to a multi-input gate increases the propagation delay through the gate [9, 18, 19]. For our purposes, this effect is only significant when tokens are spaced with $TI \leq 10$ gd. Drafting occurs over a much longer time frame. Figure 23 shows that when there is variable t_{plh} there is varying V_K and vice-versa. Near the Charlie peak there is a minimum t_{plh} on the drafting side (red arrow). This minimum corresponds to the

maximum-throughput spacing of tokens in a GasP FIFO ($TI \approx 10$ gd and $NI \approx 0$ gd). If the token spacing is slightly less than at this minimum t_{plh} , the mild anti-drafting of the Charlie effect will push them apart. If the token spacing is slightly more than at this minimum t_{plh} , drafting will pull them together. If the spacing is forced to less than the maximum throughput spacing the FIFO will no longer be operating in the sparse token region and the tokens will move based on the fixed logical delays of GasP.

2.5 The PMOS Stack Circuitry

Figure 24 shows the circuit mechanism of V_K decay during drafting. Suppose two tokens, 1 then 2, traverse the m^{th} GasP in a chain of GasPs that has been idle for a long time. When a token traverses the m^{th} GasP the NOR-gate goes from state C to A, then to B before returning to C. State A charges the K node to V_{DD} , which is preserved by state B. If token 2 follows token 1 such that $NI_m \geq 1$ gd, then in the C-state V_K decays slowly from V_{DD} to V_T through a diode-connected PMOS transistor. C-state decay of V_K is interrupted by token 2 before the K-node is fully discharged, so $t_{plh}(NI_m) < t_{plh}(\infty)$, causing the drafting effect, $TI_{m+1} < TI_m$. (TI_{m+1} is identified because $\text{Succ}_m = \text{Pred}_{m+1}$).

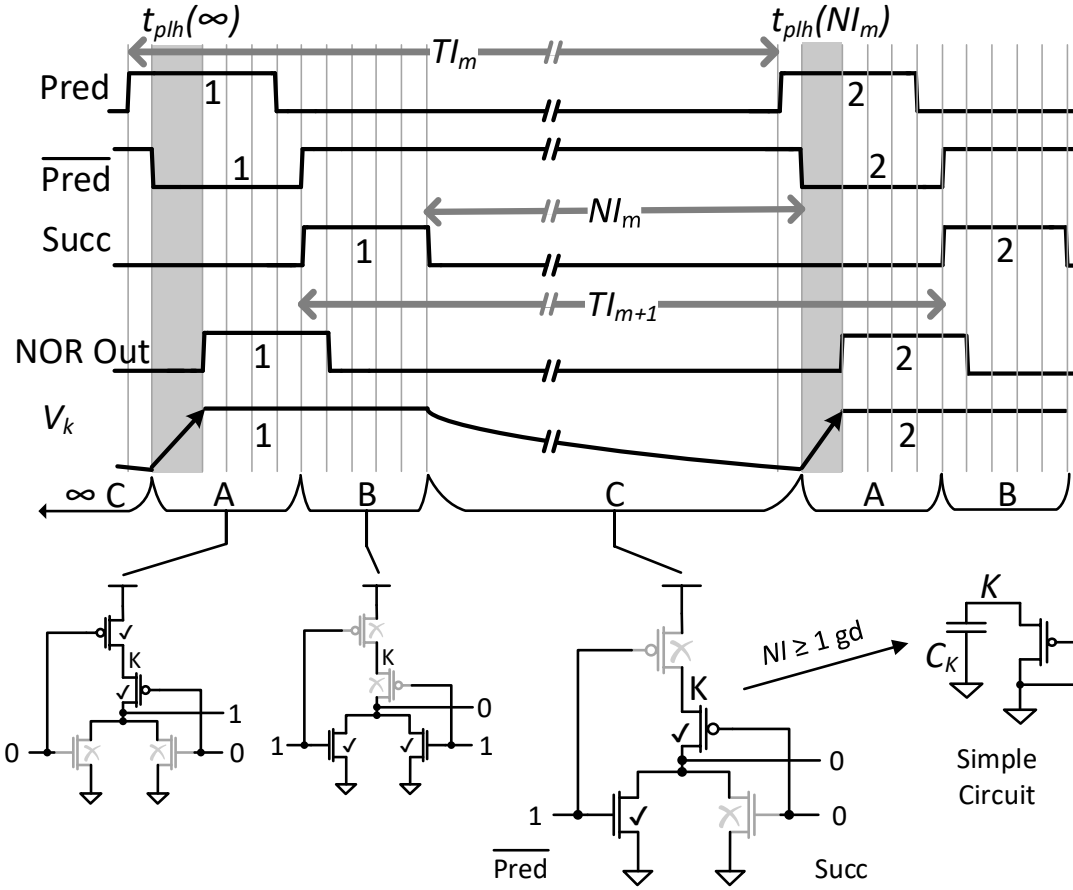


Figure 24: The NOR circuit mechanism of V_K decay. Between tokens (state C) the GasP stage is empty and waiting for a token to arrive at Pred ; Succ is logic 0 and $\overline{\text{Pred}}$ is logic 1. Node K is charged to V_{DD} and the NOR output is at logic 0. This condition results in a diode-connected PMOS transistor that discharges node K through the Pred side NMOS transistor which is turned on.

The preceding discussion was for sparsely distributed tokens propagating on a chain of GasPs which have NOR gates in the Rail Pred configuration. The other NOR configuration (Figure 14) and other FIFO workload cases can be analyzed similarly and are summarized in Figure 25. Vacancies draft with the Rail Succ NOR configuration but not with the Rail Pred configuration. Drafting of sparse tokens/vacancies occurs when the output-connected PMOS is on during the C-state, slowly discharging K . Non-drafting

of sparse tokens/vacancies occurs when the rail-connected PMOS is on during the C-state, clamping K to V_{DD} .

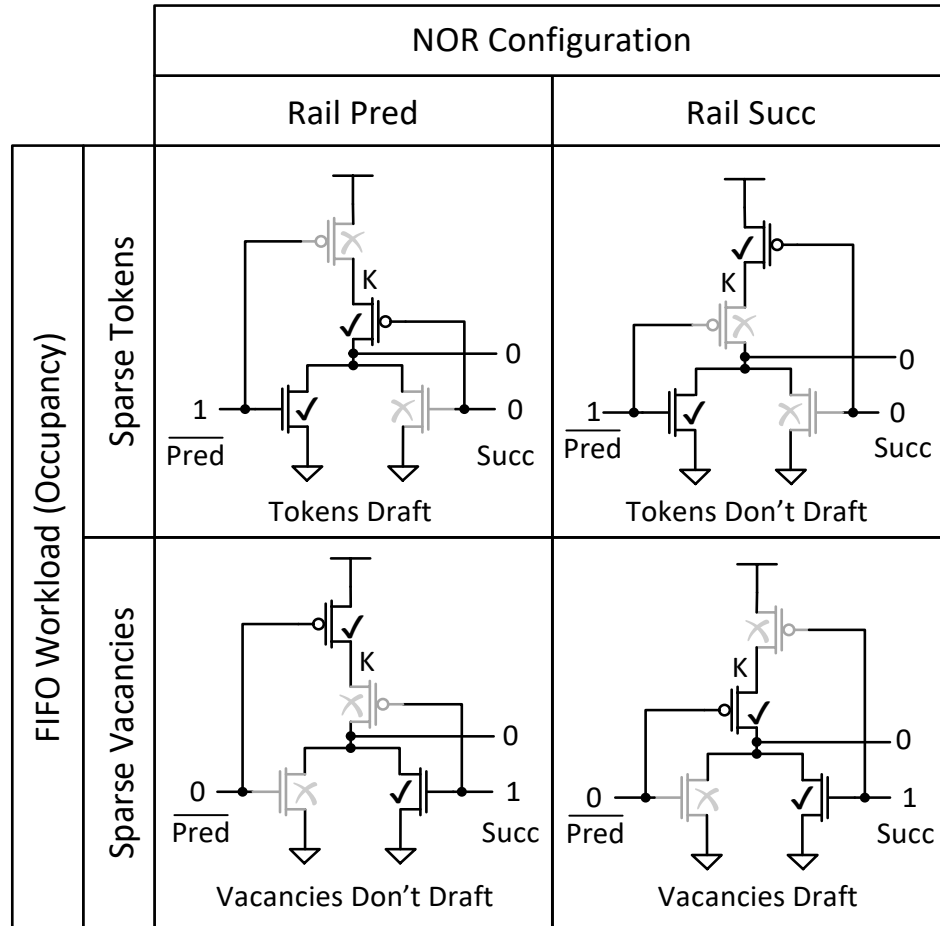


Figure 25: The GasP NOR configuration determines drafting behavior for sparse token or sparse vacancy workloads. The NOR state between tokens is shown here. Drafting occurs when the output connected PMOS is turned on before the rail PMOS.

Chapter 3 Modeling of Drafting

3.1 Deriving a K Function

Figure 24 shows the sequence of states of the NOR gate and timing of signals as two tokens traverse a Rail Pred GasP stage m in a chain of GasPs that has been idle for a long time. The most usual case of widely-spaced tokens has token 2 following token 1 by an interval, $TI_m > 10 \text{ gd}$, such that $NI_m > 0$. If logical gate delays are assumed to be unity, except that $t_{plh}(NI)$ depends on the duration NI of the preceding C-state, inspection of

Figure 24 gives Eqs. (3) and (4)

$$TI_m = 9 + NI_m + t_{plh}(\infty) \quad (3)$$

$$TI_{m+1} = 9 + NI_m + t_{plh}(NI_m) \quad (4)$$

$$TI_{m+1} = 9 + NI_{m+1} + t_{plh}(\infty) \quad (5)$$

where Eq. (5) is Eq. (3) with $m \rightarrow m + 1$. Subtraction of Eq. (5) from Eq. (4) gives

$$NI_{m+1} = NI_m + t_{plh}(NI_m) - t_{plh}(\infty). \quad (6)$$

If t_{plh} is known as a function of NI , then Eq. (6) may be iterated to find NI_m form ≥ 1 . TI_m form ≥ 1 may then be computed via Eq. (3).

A simple model of $t_{plh}(NI)$ may be derived by noticing that in the usual case of widely spaced tokens ($NI > 1 \text{ gd}$), t_{plh} is controlled by the mechanism of charge on the K-node leaking away through a diode-connected PMOS transistor during the preceding C-state, as shown in Figure 24. The complex behavior of simultaneous NOR input transitions (Charlie effect) may be modeled as simply a “stop” to drafting when $NI = 0$. For

$NI > 1$ gd in the C-state, the condition that the current discharging the K-node capacitance, C_K , flows through the diode-connected transistor is written

$$C_K \frac{dV_K}{dt} = -\kappa(V_K - V_T)^2 \quad (7)$$

where the Shichman-Hodges (SH) model of transistor action, ignoring body effects, is assumed, and where V_T is the threshold voltage. Eq. (7) can be integrated with the boundary condition that $V_K = V_{DD}$ at $t = 0$ to give

$$V_K = \frac{V_{DD} - V_T}{1 + t/\tau} + V_T \quad (8)$$

where

$$\tau = \frac{C_K}{\kappa(V_{DD} - V_T)}. \quad (9)$$

If we assume, per Figure 22b, that t_{plh} is a linear function of V_K , that $V_K = V_T$ corresponds to $t_{plh}(\infty)$, and $V_K = V_{DD}$ corresponds to $t_{plh}(0)$, then the simplest model of $t_{plh}(V_K)$ is

$$t_{plh}(V_K) = \left(t_{plh}(0) - t_{plh}(\infty) \right) \frac{V_K - V_T}{V_{DD} - V_T} + t_{plh}(\infty) \quad (10)$$

or, using Eq. (8)

$$\begin{aligned} t_{plh}(t) &= -\Delta t_{plh} \frac{\tau}{\tau + t} + t_{plh}(\infty) \\ \Delta t_{plh} &= t_{plh}(\infty) - t_{plh}(0). \end{aligned} \quad (11)$$

If t in Eq. (11) is identified as NI , then the parameters $t_{plh}(0)$, $t_{plh}(\infty)$, and τ may be extracted from a fit to t_{plh} vs NI measured data from a Rail Pred NOR such as shown in Figure 26. Fitted values of the parameters are given in the figure.

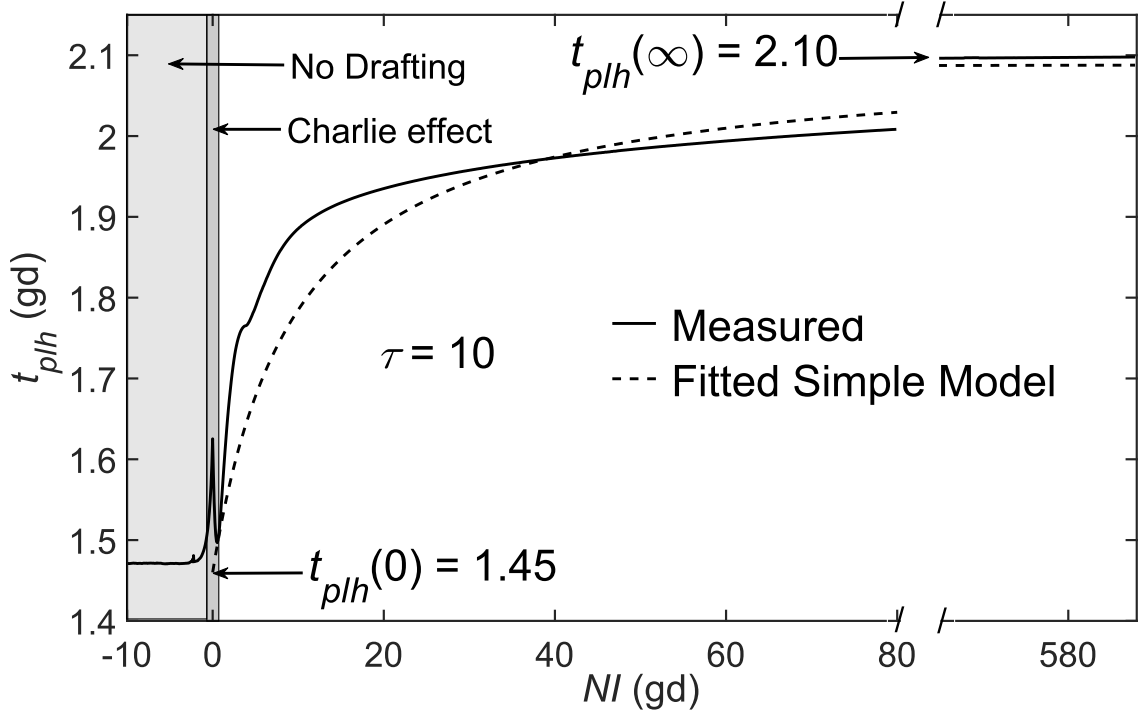


Figure 26: NOR propagation delay t_{plh} vs NOR input arrival time difference NI for the Rail Pred configuration. t_{plh} is a strong function of NI for $NI \geq 1$ gd but is nearly independent of NI for $NI \leq 1$ gd. For $NI \geq 1$ gd the positive slope of $t_{plh}(NI)$ causes drafting, for which a simple model (dashed line) can be derived. A token will stop drafting when the interval to a preceding token falls to $TI \approx 11$ gd ($NI \approx 1$ gd) and the Charlie effect stops further drafting.

Substitution of Eq. (11) into Eq. (6) gives

$$NI_{m+1} = NI_m - \Delta t_{plh} \frac{\tau}{\tau + NI_m} \quad (12)$$

which may be iterated to find NI_m , $m \geq 1$. The continuum limit of Eq. (12) is easily integrated to give a closed-form formula. The approximation is good because the change in NI per GasP is usually small ($NI_m - NI_{m+1} \ll \Delta t_{plh} = 2.1 - 1.45 = 0.65$ gd). The continuum limit is written

$$NI_{m+1} - NI_m \rightarrow \frac{dNI}{dm} = -\Delta t_{plh} \frac{\tau}{\tau + NI}, \quad (13)$$

which integrates to a closed form

$$NI_m = \left((NI_0 + \tau)^2 - 2\tau\Delta t_{plh}m \right)^{1/2} - \tau. \quad (14)$$

Figure 27 shows the NOR input interval NI_m resulting from two tokens traversing GasP stage m in a Rail Pred GasP FIFO, for $m \geq 0$. Token 2 is injected onto GasP element $m = 0$ following token 1 by $TI = 50$ gd so that $NI_0 = 40$ gd. The FIFO uses the parameters of the example shown in Figure 26. SPICE simulation (Figure 27) shows that the second token catches up with the first and becomes fully drafted at GasP stage $m = 178$. Also given in Figure 27 are results of using Δt_{plh} and τ from Figure 26 to iterate the simple model, Eq. (12), and to evaluate the closed form approximation, Eq. (14). The good agreement between the SPICE simulation, the simple model, and closed form shows that the latter are useful approximations. The number of stages, m , required for a pair of tokens separated initially by NI_0 to fully draft is the solution to Eq. (14) with $NI_m = 0$. That is

$$m = \left((NI_0 + \tau)^2 - \tau^2 \right) / (2\tau\Delta t_{plh}) \quad (15)$$

which, for this example, gives $m = 185$ stages.

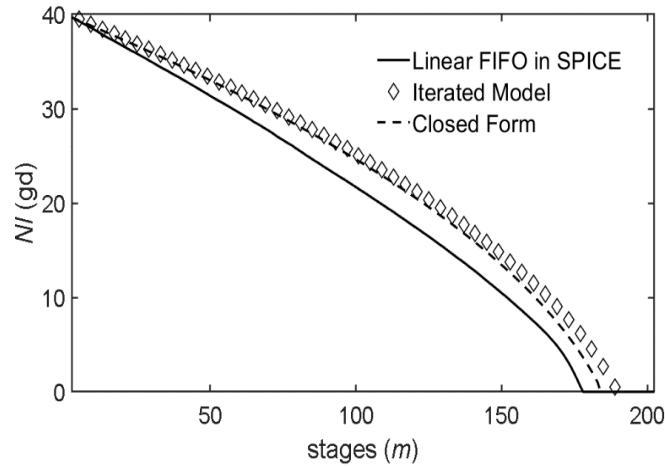


Figure 27: SPICE and drafting model predictions for drafting of two tokens are compared. The iterated SH model, the closed form SH model, and a linear FIFO simulated in SPICE agree. The starting NI for each is 40 gd.

3.2 Behavior of the Simplified Circuit

The measured K-node behavior of the NOR gate was compared with two simplifications of the NOR gate and with the Simplified Model (SH model described above). The two simplifications of the NOR gate were the PMOS stack and a further simplification (“Simple Circuit”) shown in Figure 28. Both were simulated in SPICE. Good agreement among of V_K vs NI for measured K-node behavior and the two simplifications is apparent in Figure 28. However, the simple Shichman-Hodges (SH) model, Eqs. (7) and (8), plotted using parameters given in Figure 27 deviates from the others. The deviation shows that the body effect on threshold voltage (V_T), ignored in the SH model, is quite significant. I_D does not follow the SH model when the V_K falls below $V_{DD}/2$ because the diode-connected PMOS transistor moves into sub-threshold conduction (see Section 3.3). The K node parasitic capacitance C_K includes the diffusion capacitance at the junction of the drain of the upper rail-connected PMOS and the source capacitance of the lower

output-connected PMOS transistor. This node was mentioned by previous authors but was not fully investigated [16, 17]. The NOR gate is the only combinational circuit in GasP that contains a node which is sometimes isolated from active drive and can therefore drift over a very long time between NOR actions.

The K node parasitic capacitance C_K also decreases as V_K drops because of increasing reverse bias across the n-well/source-drain diffusions, but the agreement of the “Simple Circuit” with the NOR and PMOS stack shows that the effect is small. Although the SH model omits the significant effect of sub-threshold conduction and body bias effects, it is mathematically simple, Eqs. (12), (14), and (15), and gives qualitatively useful results (Figure 27).

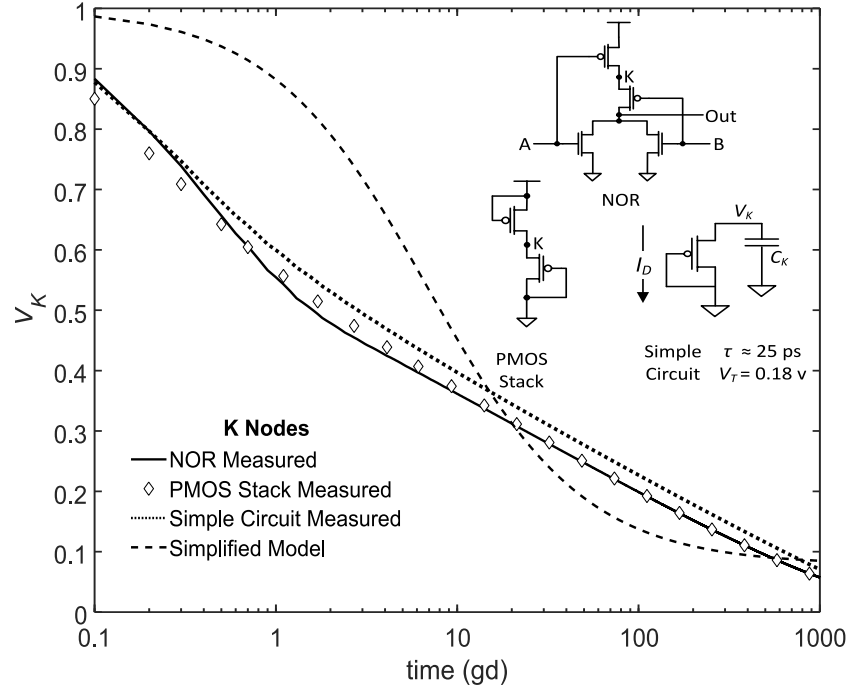


Figure 28: Fitting the simplified model for V_K to data from HSPICE simulation. Note that the model curve departs from the measured data. This is due to the body effect which raises V_T and decreases C_K as V_K decreases.

3.3 The Body Effect on V_K

Subthreshold conduction and extreme body effects are not directly accessible in SPICE models, but threshold voltage can be measured in SPICE simulation. Figure 29 shows a SPICE test bench for measuring this body effect for the NOR PMOS transistor.

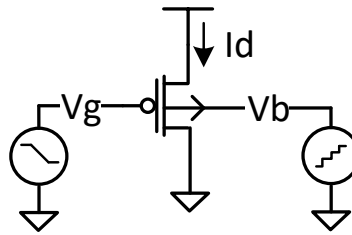


Figure 29: PMOS test Bench.

Body voltage (V_b) is stepped and gate voltage (V_g) is swept from V_{DD} to 0 V for each step. Drain current (I_D) is measured during the sweep of V_g , for each step in V_b . Body voltage effects are shown in Figure 30. I_D is shown both in linear and log scales and V_g is shown in absolute value for simplicity. Note that with a body voltage bias greater than 0.5 V there is significant leakage due to Drain Induced Barrier Lowering (DIBL).

For each V_b , the square root of I_D can be plotted against V_g and the linear portion extrapolated to the x -axis to get V_T (Figure 31, left). V_T changes only slightly with V_b until V_b is greater than 0.5 V beyond which there is no V_T because the body is acting as a back gate and the transistor is conducting. Since $V_K = V_{DD} - V_b$, V_T can be plotted against V_K which shows the marked departure of V_T when V_K drops to half supply, 0.5 V (Figure 31, right). V_K below $V_{DD}/2$ forces the transistor into sub-threshold conduction.

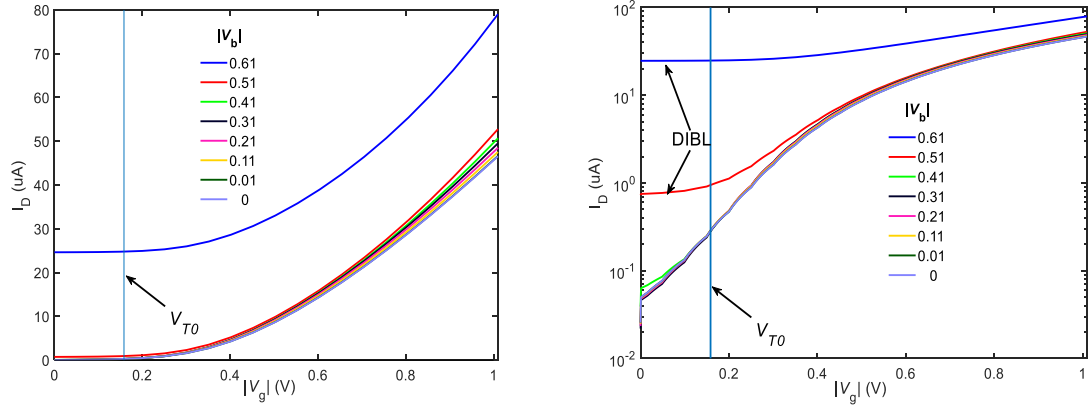


Figure 30: PMOS current (I_D) vs gate voltage (V_g) for different body voltages (V_b). When the body voltage reaches 0.5 V the transistor starts to conduct via DIBL.

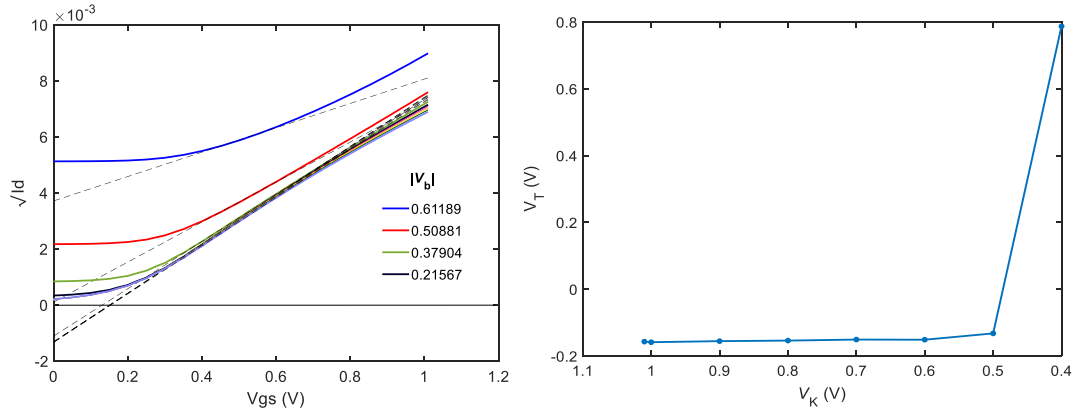


Figure 31: Threshold voltage (V_T) and V_K . (left) Extrapolating $\sqrt{I_d}$ to get threshold voltage for different body voltages. (right) Threshold voltage (V_T) vs. V_K . At $V_K < V_{DD}/2$ there is marked change in V_T .

The K function decays very rapidly to $V_{DD}/2$ (Figure 15, top) which means that almost all the drafting behavior occurs with the diode connected PMOS transistor in subthreshold conduction. Substituting the expression for subthreshold current for the Shichman-Hodges (SH) model in Eq (7) makes model derivation intractable.

3.4 MATLAB Modeling of Drafting

The essential role of the K function and the variable t_{plh} of the NOR in drafting can also be shown by abstracting the GasP FIFO operation into a MATLAB simulation. Two types of simulation were performed. The first is an event-driven simulation where each gate in GasP is replaced with a timer, and the second is a simpler interval-based simulation.

For the event-driven simulation each of the timers has two delays. One delay is the forward propagation of HIGH to LOW (t_{phl}) and the other delay is the forward propagation of LOW to HIGH (t_{plh}). All gates, except the NOR, have fixed t_{phl} and t_{plh} delays with the same nominal value, made to have the same delays as the 32 nm SPICE circuit. The NOR gate has a t_{phl} delay with the nominal value, but it has a variable t_{plh} delay which is a function of TI . The t_{plh} of the NOR gate vs. TI was measured from a 32 nm SPICE simulation of a GasP FIFO. This is then used as a lookup function in MATLAB. The NOR and the central buffers of the GasP stage are incorporated into a combined, but variable, C-timer. The P timer represents the inverter that creates $\overline{\text{Pred}}$, the PC timer is the Pred state-wire driver and the S timer is the combined delay of the Succ state-wire driver and its driving inverter (see Figure 32).

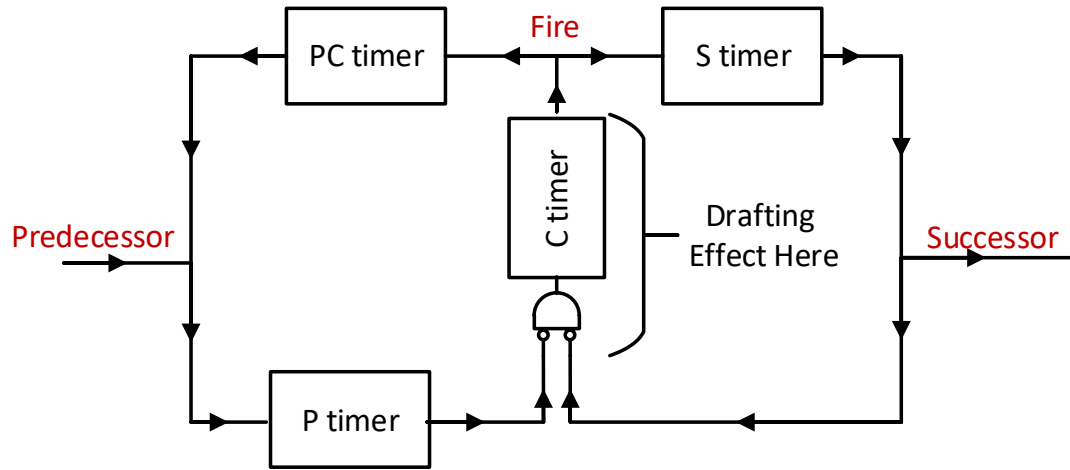


Figure 32: The MATLAB event-driven model of GasP. The C timer is variable modelling the NOR gate.

An interval-based simulation is simpler and much faster than the event-driven simulation because it only looks at the changes in the intervals between tokens. Here the concern is with the interval sizes and not in the speed of propagation of the tokens through the FIFO. Focusing only on the intervals captures the effects of the K function without unnecessary complexity.

Figure 33 shows three tokens passing through one stage on a ring FIFO. Changes to intervals that occur when tokens 1, 2, and 3 pass through a GasP stage are shown in successive “snapshots” (top to bottom) in the figure. Note that when an interval is shortened by an amount $K(*)$ the next interval must be lengthened by the same amount. This is because tokens are not affected by changes in a stage until they reach that stage. For instance, as token 1 passes through the stage it gains on token 3 by time $K(C_L)$, shortening interval C_L to $C_N = C_L - K(C_L)$. Token 2 is yet unaffected by the stage but now follows token 1 by a longer interval $A_L = A + K(C_L)$. When it arrives at the stage it will then be shortened by an amount, $K(A_L)$, based on the longer interval, A_L .

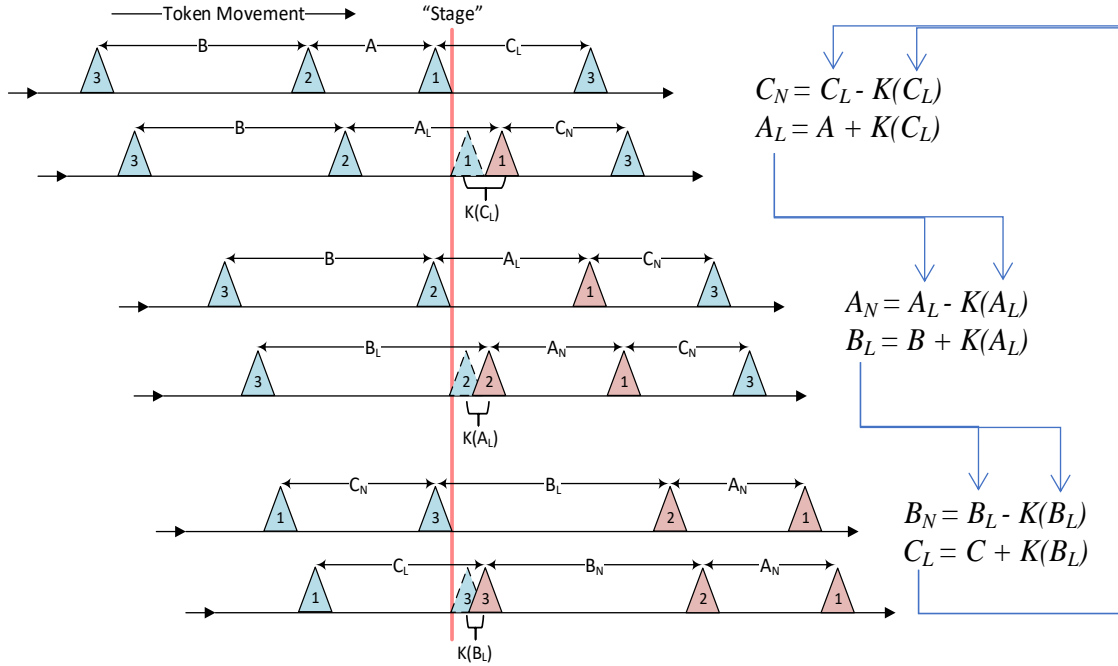


Figure 33: The difference equations for three drafting tokens in a ring FIFO. Three successive views showing changes in intervals when tokens 1, 2, and 3 pass through a GasP stage are shown as “snapshots” (top to bottom). Corresponding difference equations for intervals and the method of iteration are shown.

A set of difference equations describing changes of intervals as tokens pass a GasP stage may be written down, and the equations may be iterated, $\{A_N, B_N, C_N\} \leftarrow \{A_L, B_L, C_L\}$, as shown in Figure 33. If any new interval is calculated to be less than the minimum interval, which is measured in SPICE and comes from the cycle time of the GasP stages, the new interval is shortened only to minimum and that amount added to the trailing interval. This mimics what happens in the actual circuit.

The 17 stage SPICE FIFO was modelled in MATLAB as an event driven simulation and as an interval-based simulation. The same starting conditions were used for all three and the NOR t_{plh} look-up function, derived from SPICE, was used in the MATLAB models.

Figure 34 shows the comparison of SPICE and both MATLAB models. There is good agreement between all three simulations. This shows that the abstractions in MATLAB faithfully captures the drafting behavior in SPICE and that the K node and its decay function are the essence of drafting.

Note that the intervals cannot move completely into the apex of the ternary diagram because intervals can never be zero. The divisions on the ternary graph are stage delays through GasP which is 6 gate delays. There are 17 stages, so the maximum possible interval is 17 units. The minimum spacing for GasP is 10 gate delays which is 1.6 stage delays. With full drafting all but one interval is minimum, and the plotted location is 1.6 units from the ternary apex.

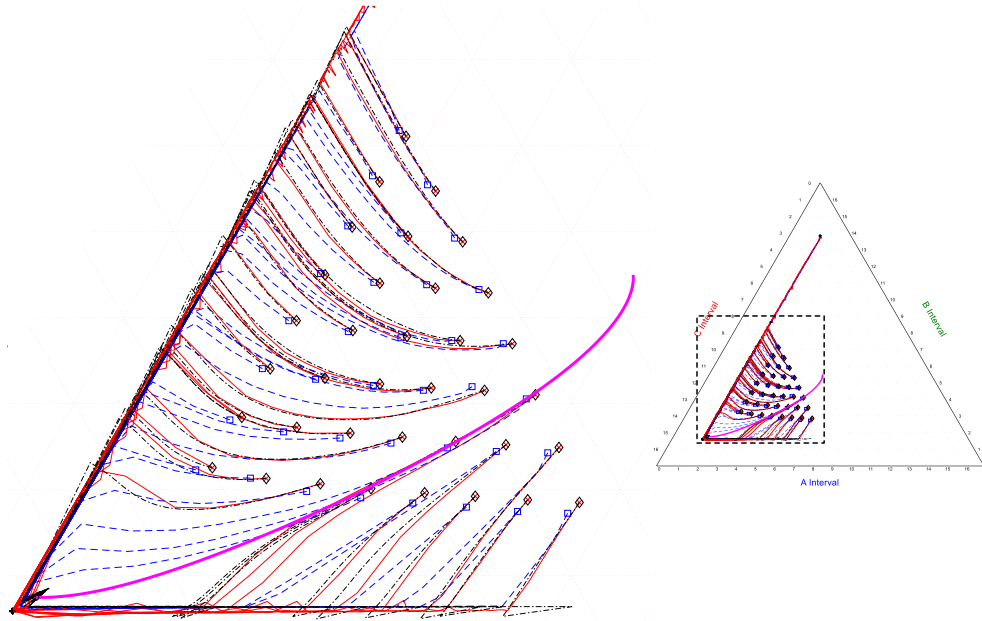


Figure 34: SPICE and MATLAB simulations on a ternary graph. Shown are SPICE simulations (continuous red lines), MATLAB event-driven simulations (dashed blue lines), and MATLAB interval-based simulations (black dash-dot line). Many starting token interval combinations were simulated. Left is a magnified inset of the right plot. The solid magenta line is described in Section 3.5

In the SPICE simulations tokens can be initialized by pre-setting the state-wires but this offers only coarse initial intervals at multiples of stage delays. For finer initial interval settings, as needed for Figure 34, a token injection technique was used. There is one special GasP stage in the FIFO that contains extra circuitry (Figure 35) to place 3 tokens on the FIFO. The successor is initialized with a single token. The simulation begins, and this single token circulates through the rest of the FIFO and arrives back at this stage. A window is set to capture the token which is then delayed and reinjected twice on the successor. This creates three tokens with more precise timing. The inter-token intervals are set by delay 1, (interval between tokens 1 and 2) and delay 2 (interval between tokens 2 and 3).

The same precision can be achieved in the event driven MATLAB model by presetting the gate counters in the GasP stages (Figure 32). In SPICE, and in real circuits, tokens do not “pop” from predecessor to successor but rather propagate through the stages’ gates smoothly. In MATLAB event driven simulation, presetting a stage’s counters will position the transiting token more precisely in that stage at start-up.

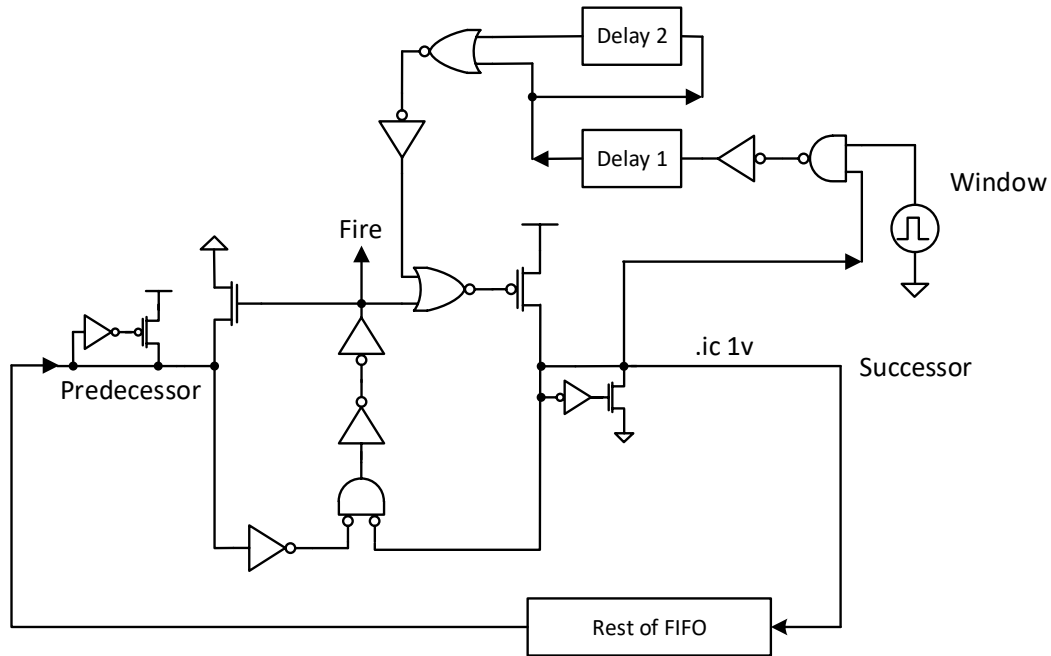


Figure 35: Technique for injecting 3 tokens at precise intervals. An initial token placed on this stage's successor circulates through the rest of the FIFO after start-up and is reinjected two more times as the simulation continues. This causes two more tokens to be injected for a total of three. Inter-token intervals are determined by delay 1 and delay 2.

3.5 Line of Demarcation

An interesting result appears in the ternary graph, Figure 34, of trajectories with multiple starting interval combinations. A combination with two smaller starting intervals, the first slightly smaller than the second, might be expected to draft with the first, smaller, interval collapsing before the second, larger interval. But this does not happen. There is a line of demarcation which is the watershed between the fates of two sequential intervals in a three-token circular FIFO. One might expect this watershed to be $A = B$ line but it is not. It also appears that a larger first interval can collapse first. This can be explained by noting that the first token sees the longest preceding interval and therefore propagates the

slowest of all three. In effect, the first token “backs up” into the second faster than the second backs up into the third. The line of demarcation can be derived as follows:

Given three intervals on a circular FIFO in the order of A-B C, consider when will interval A collapse first even if it is larger than interval B (Figure 36). If the fraction shortening of A, Δ_2 , is more than the fraction shortening of B, Δ_3 , then A will collapse before B.

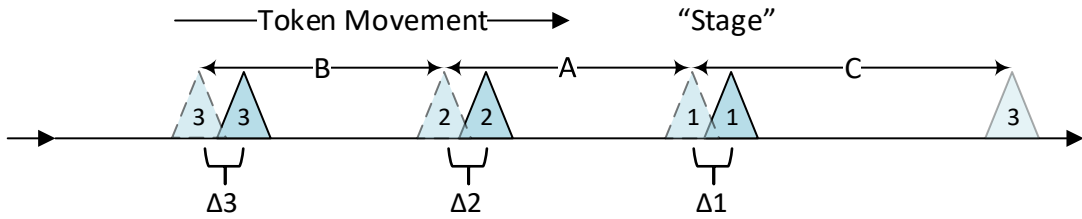


Figure 36: Interval algebra to find the line of demarcation.

This occurs if: $\frac{(\Delta_2 - \Delta_1)}{A} > \frac{(\Delta_3 - \Delta_2)}{B}$. To generate the line of demarcation the boundary condition $\frac{(\Delta_2 - \Delta_1)}{A} = \frac{(\Delta_3 - \Delta_2)}{B}$ is used.

Substituting the SH model, Eq. (8), for each delta, $\Delta = \frac{b}{1+aX}$, where a and b are

constants and X is the interval preceding a given token gives the boundary. For instance,

token 1 moves toward token 3 by interval $\Delta_1 = \frac{b}{1+aC}$ because interval C precedes token

1. This shortens interval C and lengthens interval A by Δ_1 . Interval A is now $A + \Delta_1$ or A^* . Since the Δ s are added and subtracted sequentially, the boundary condition is now:

$$\frac{\left(\frac{b}{1+aA^*} - \frac{b}{1+aC}\right)}{A} = \frac{\left(\frac{b}{1+aB^*} - \frac{b}{1+aA^*}\right)}{B} \quad (16)$$

Because this is a circular FIFO there is an additional constraint that $C = L - (A + B)$

where L is the combined length of A, B and C

$$\frac{\left(\frac{b}{1+aA^*} - \frac{b}{1+a(L-A-B)}\right)}{A} = \frac{\left(\frac{b}{1+aB^*} - \frac{b}{1+aA^*}\right)}{B} \quad (17)$$

Substituting the Δ 's gives a relatively intractable equation in A and B.

$$\frac{\frac{b}{1+a\left(A+\frac{b}{1+aC}\right)} - \frac{b}{1+a(L-A-B)}}{A} = \frac{\frac{b}{1+aB+\frac{b}{1+a\left(\frac{b}{1+aC}\right)}} - \frac{b}{1+a\left(A+\frac{b}{1+aC}\right)}}{B} \quad (18)$$

This can be solved using a combination of MATLAB algebra and is plotted as the magenta line in Figure 34. The derivation is tested against multiple starting interval combinations in the MATLAB interval simulation. Figure 37 shows the simulations in black and the previous magenta line of demarcation. The magenta line derived from Eq. (18) deviates slightly from the watershed apparent in the MATLAB-computed trajectories (black) especially for smaller intervals. The reason is that the K function used for the MATLAB trajectories (SPICE-based lookup) is slightly different from the K function used in Eq. (18) (SH model-based). Still, there is general agreement confirming that the first interval (A) can collapse first, even if it is larger than second interval (B).

Figure 38 shows one interval combination that demonstrates early A collapse even though at the start (S), $A > B$.

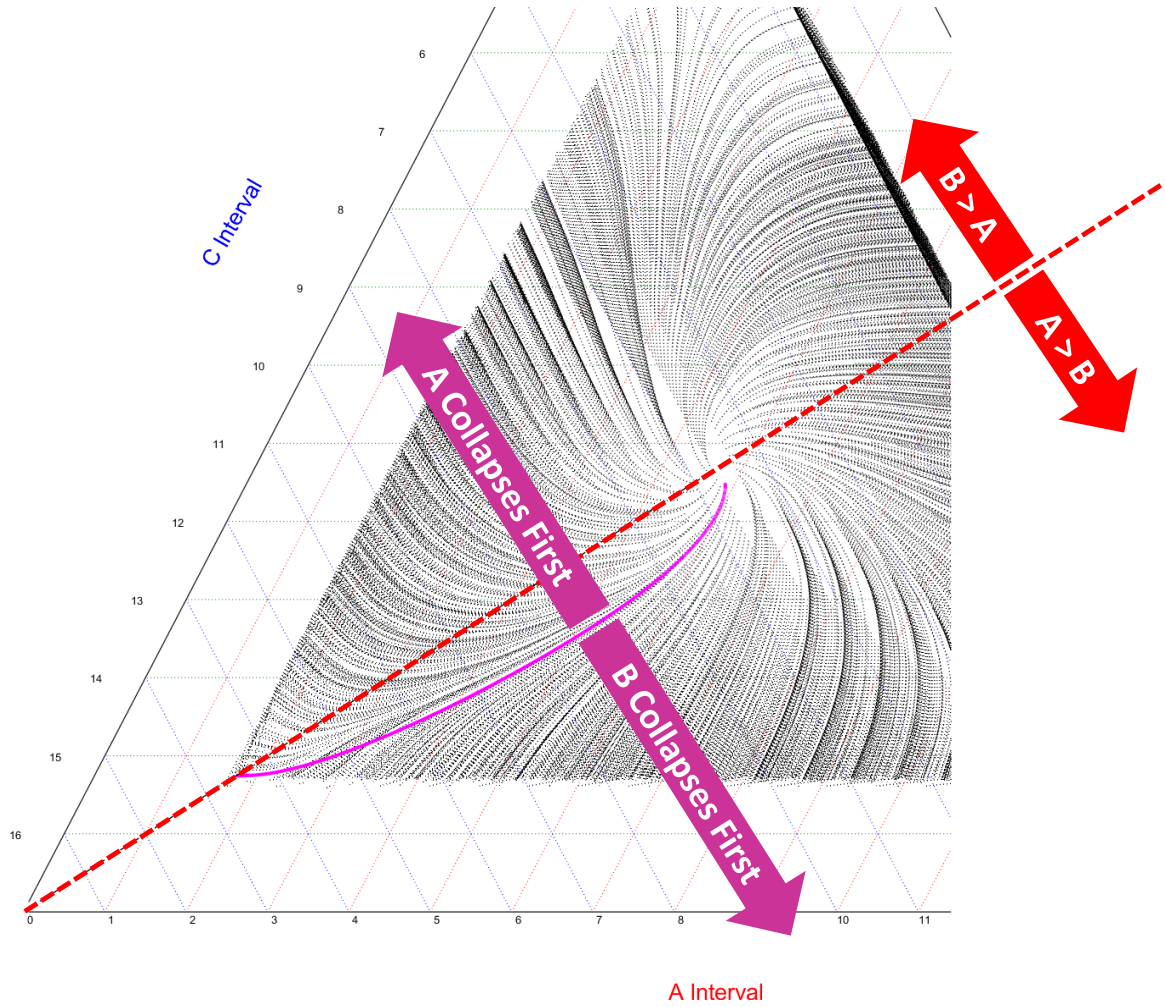


Figure 37: Line of demarcation demonstrated with multiple MATLAB simulations. Dotted line is $A = B$ and magenta line is the line of demarcation derived in Section 3.5. Sometimes even the longest interval (A) will collapse first.

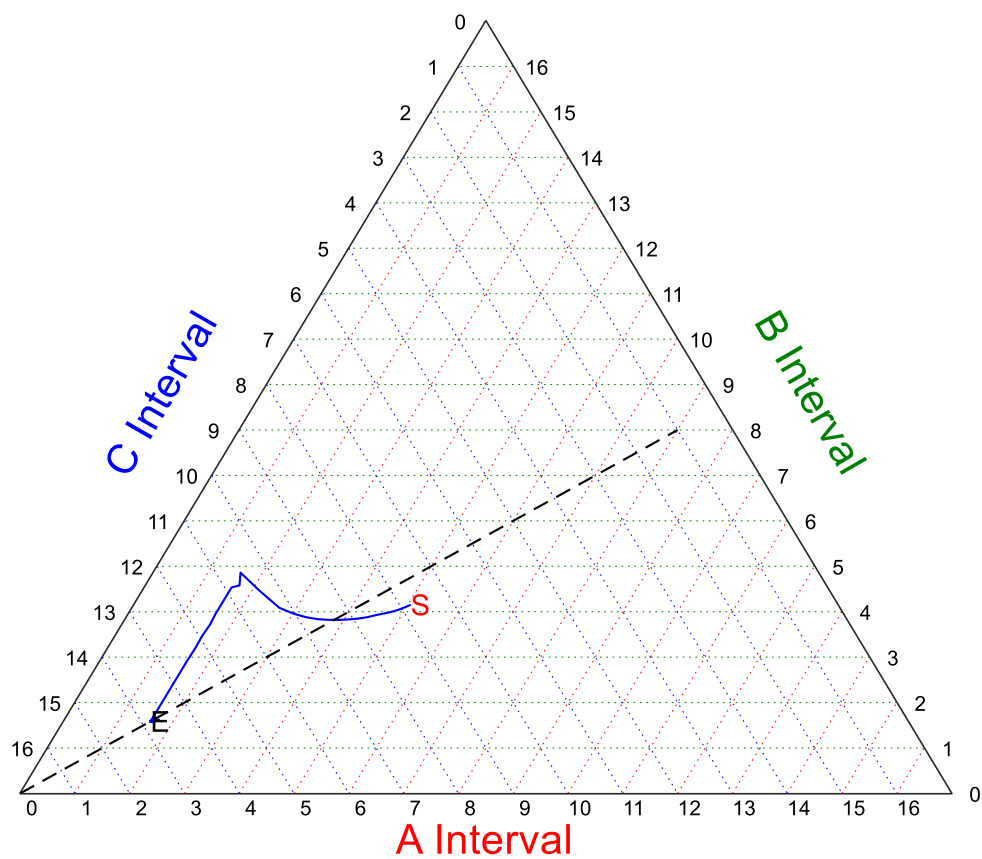


Figure 38: One example of longer first interval collapsing first on a ternary graph. At "S" the A interval (first) is greater than the B interval (second) but A collapses first. Dashed line is $A = B$.

Chapter 4 A Control Circuit

Because the decay of V_K between tokens causes token drafting, one expects that a rising V_K between tokens should generate anti-drafting. Also, because a Rail Succ configuration clamps V_K to V_{DD} preventing drafting, holding V_K constant between tokens in a Rail Pred configuration is expected to prevent drafting or anti-drafting regardless of the token separation. Each time the GasP stage fires and moves a token to the output state-wire, node K is reset. For drafting, the control circuit does not affect the normal reset to V_{DD} or V_K decay inherent in the NOR. For anti-drafting the control circuit forces a reset to 0 V, overriding the normal NOR reset, and then creates a rising V_K . To prevent drafting or anti-drafting, node K is reset to $V_{DD}/2$ and clamped there.

Figure 40 schematically shows the GasP NOR and the desired waveforms to control the voltage on the K node. Each GasP stage in our test FIFO has a copy of the control circuit that connects directly to the Rail Pred NOR K node in that stage and can alter its voltage over time, over-riding the normal decay of V_K . The control circuit uses that stage's fire signal to modify V_K at the correct time to preserve the logical function of the NOR. Node K is controlled only when it is not actively driven which is when the GasP stage is empty and waiting for a new token to arrive. This condition occurs between the falling edge of Succ and falling edge of $\overline{\text{Pred}}$. The control inputs to each copy of the control circuit are NO (no drafting or anti-drafting) and AD (anti-drafting). These inputs are mutually exclusive and are asserted with a logic high. Fire is the GasP Fire signal for that stage.

Each stage could be designed for independent control but for this design, all stages in the FIFO are controlled in parallel by common NO and AD inputs.

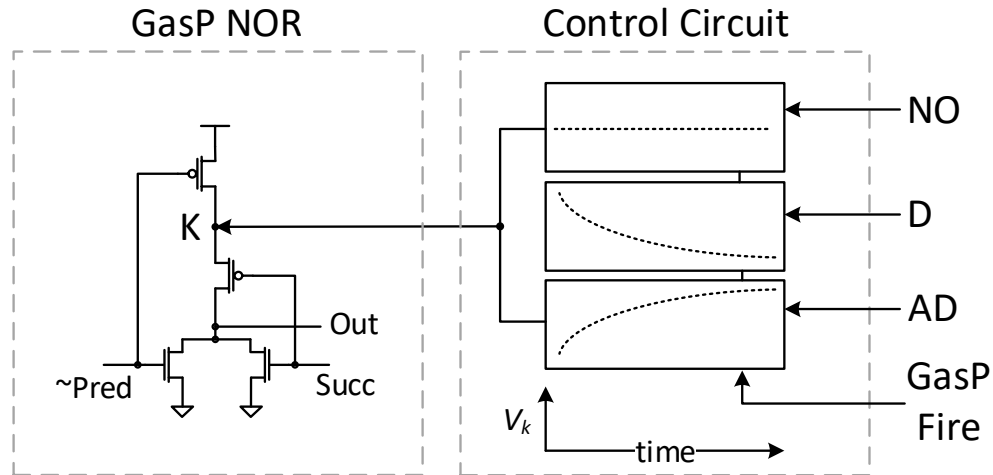


Figure 39: The intended effect of the control circuit.

The circuit details are shown in Figure 40. For anti-drafting, AD is logic high and the Fire signal turns on N1 which discharges node K and turns off the NOR. Because the Fire input propagates through three gates in the control circuit before discharging K the discharge will happen about the same time as when the Succ line is driven logic high which also turns off the NOR in the native GasP circuit. Thus, normal NOR logic function will be retained. Between Fire signals node K will then charge through P2 and the diode-connected P1. This yields a long charging time constant which is like the long discharging time constant with drafting.

For no drafting or anti-drafting, NO is asserted, and node K is clamped to $V_{DD}/2$ by P3 between firings. This voltage was selected because is a good average of the drafting and anti-drafting V_K values. Any fixed voltage from 0 V to V_{DD} will work.

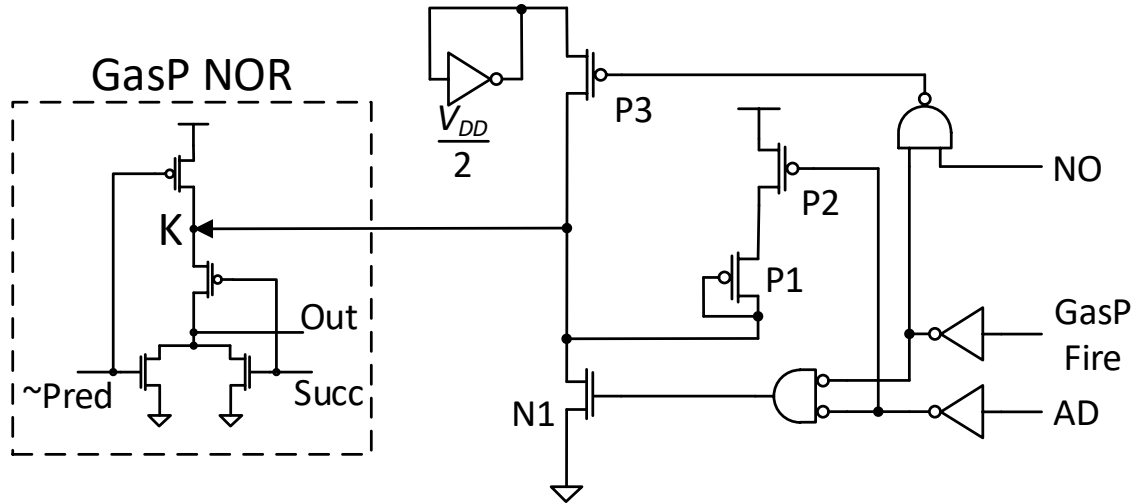


Figure 40: Details of the control circuit for node K. For drafting node K is unaffected. For anti-drafting node K is discharged by P1 and then charged by P1 and P2. For no drafting or anti-drafting node K is clamped by P3 to $V_{DD}/2$, here generated by the shorted inverter which stabilizes at this voltage.

Figure 41 shows the waveforms as measured at one stage in the FIFO during simulation with the control circuit at each stage. During normal drafting (D) there is the usual decaying function of V_K associated with drafting. During AD the V_K waveform rises which gives the inverse action on the NOR t_{plh} and anti-drafting behavior. During NO V_K remains constant between firings and there is neither drafting nor anti-drafting. These SPICE measurements show that the control circuit creates the desired waveforms for V_K .

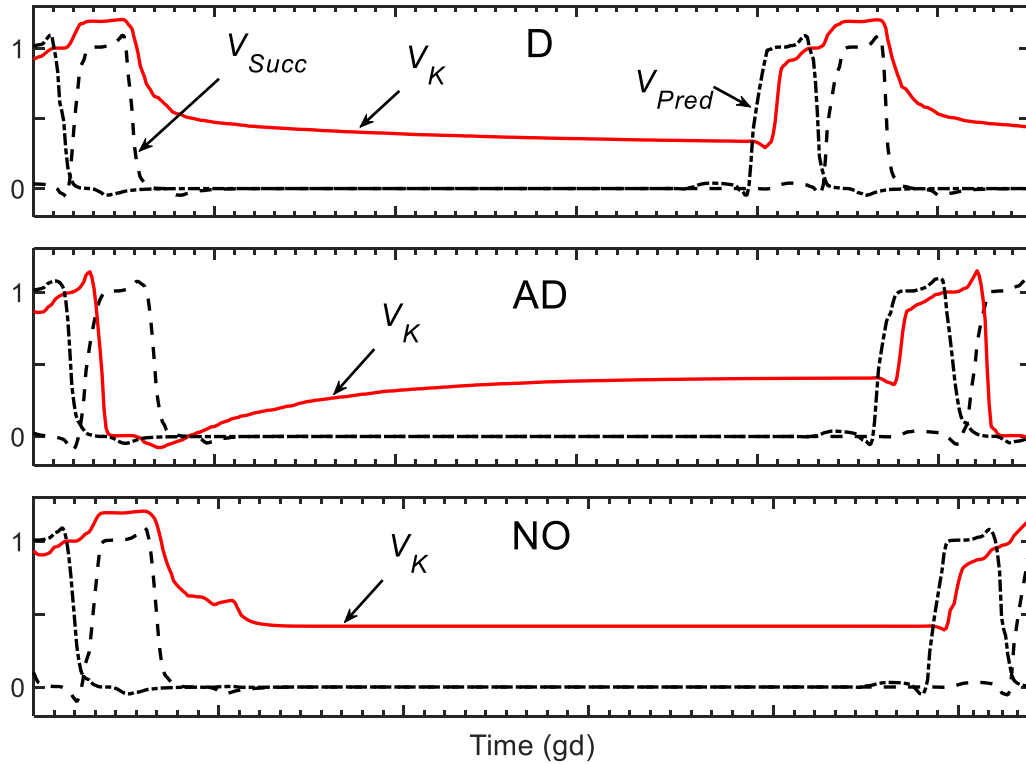


Figure 41: Node K waveforms generated by the control circuit.

Figure 42 shows how the control circuit functions while the FIFO is in operation. Three tokens are placed in a 17 stage Rail Pred circular FIFO with control circuits in each stage. The arbitrary starting positions (stages 1, 4 and 9) gives A, B and C starting IFs of 5/17, 3/17 and 9/17 of a total ring delay, respectively. IF A is between the first and second token, IF B between the second and the third and IF C between the third and the first as the first recirculates. The control circuits were exercised while the tokens circulated with a sequence of commands NO, D, NO, AD and NO. Command D means the control circuit is disabled. The control signals were asserted as indicated by arrows in Figure 42. During NO, no drafting or anti-drafting occurs, and the IFs do not change. During D

normal drafting happens and the tokens become totally drafted after 1900 gate delays. During AD the tokens move to equal-spacing which is fully anti-drafted. The control circuit shows how direct manipulation of the V_K profile provides control of all drafting and anti-drafting behavior.

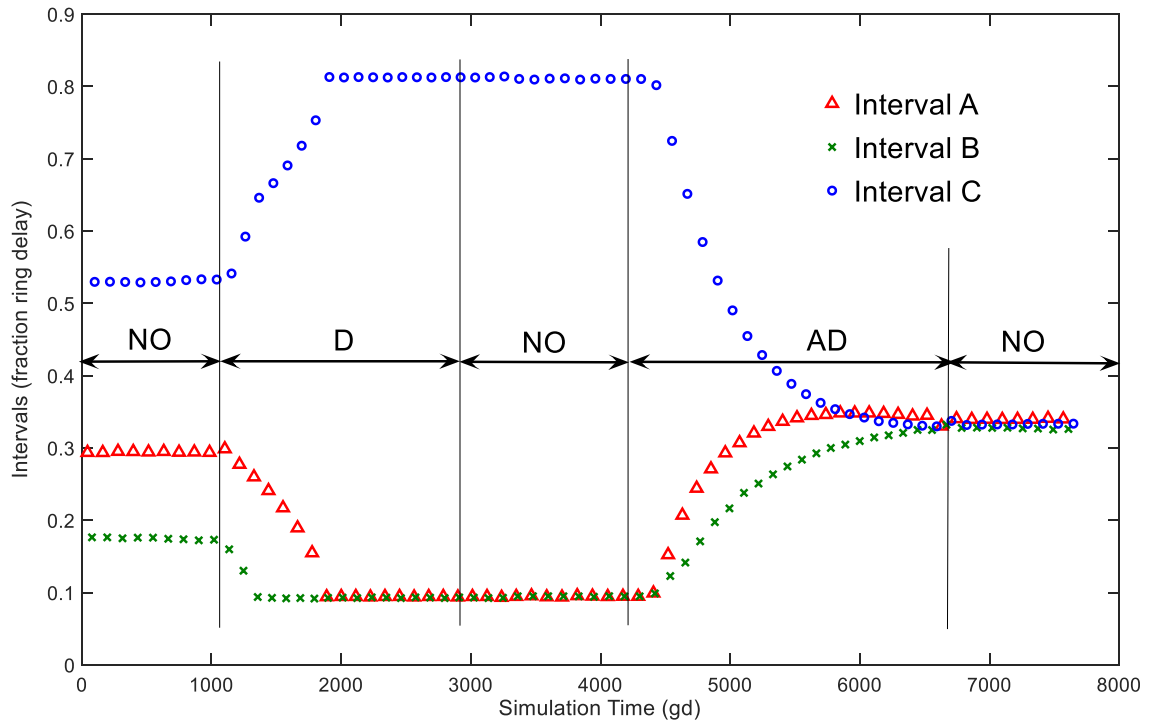


Figure 42: Control circuit results. During NO (no drafting or anti-drafting), the intervals do not change. During D (drafting) the intervals progress into the usual drafted pattern (see Figure 8). During AD (anti-drafting) the usual equal spacing of fully anti-drafted occurs.

Chapter 5 Drafting with Different Technologies and Designs

5.1 GasP NOR Gate t_{plh} vs NI in Different Technologies

The NOR test bench described in Figure 21 was used to measure the NOR t_{plh} vs NOR interval (NI) of the GasP NOR gate in other technologies. The 32 nm Synopsys model used in most of this work is compared to the TSMC 90 nm and 180 nm models in Figure 43. As the feature size increases there is still the rising t_{plh} with positive NOR interval (NI) characteristic of drafting, albeit with proportionately decreasing variation with NI . FIFO drafting has been observed in SPICE simulation of GasP FIFOs in all three technologies.

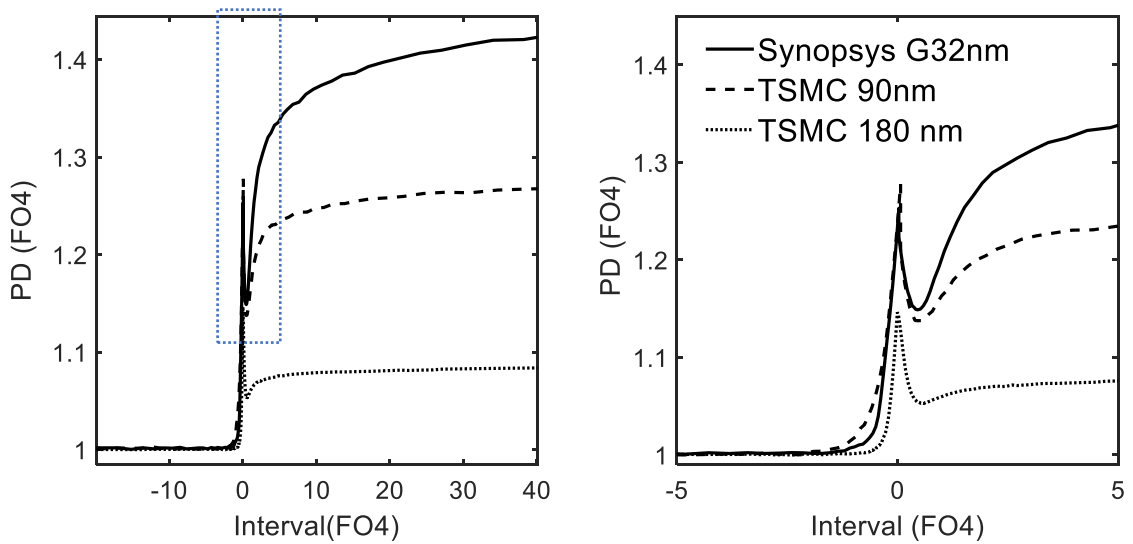


Figure 43: Comparing t_{plh} vs NOR Interval (NI) in three technologies. The right is a magnified view of the left. As feature size increases so do the propagation delays but the essential shape of the curve is maintained.

5.2 Other Self-Timed Designs.

The NOR test bench can be employed to evaluate other decision gates used in other self-timed FIFO designs such as Click, Mousetrap and Micropipeline. Also, different NOR implementations such as the symmetric NOR, SNOR, could be used for the GasP-based FIFO. Figure 44 shows four decision gate designs different from the conventional NOR design discussed so far (Figure 5). The propagation delay (t_p) vs NI characteristic of the four kinds of gates was extracted for the 32 nm technology node using a test bench like Figure 21. Gates in all examples were driven and loaded by FO4. Results of the simulations are shown in Figure 45. All show the rising t_p with increasing positive input interval characteristic of drafting. All gates have at least one internal node that cause drafting. Depending on how the gate is used in the design, these nodes could drift between tokens. FIFO designs which use these decision gates will draft by the same mechanism as GasP drafts with the NOR; changing t_p with changing input interval. Notice that GasP using SNOR will draft for *all* of the cases shown in Figure 25. FIFOs based on Micropipeline, Click, and Mousetrap were simulated in SPICE, to clarify the behavior of the gates, other than the SNOR, in Figure 44.

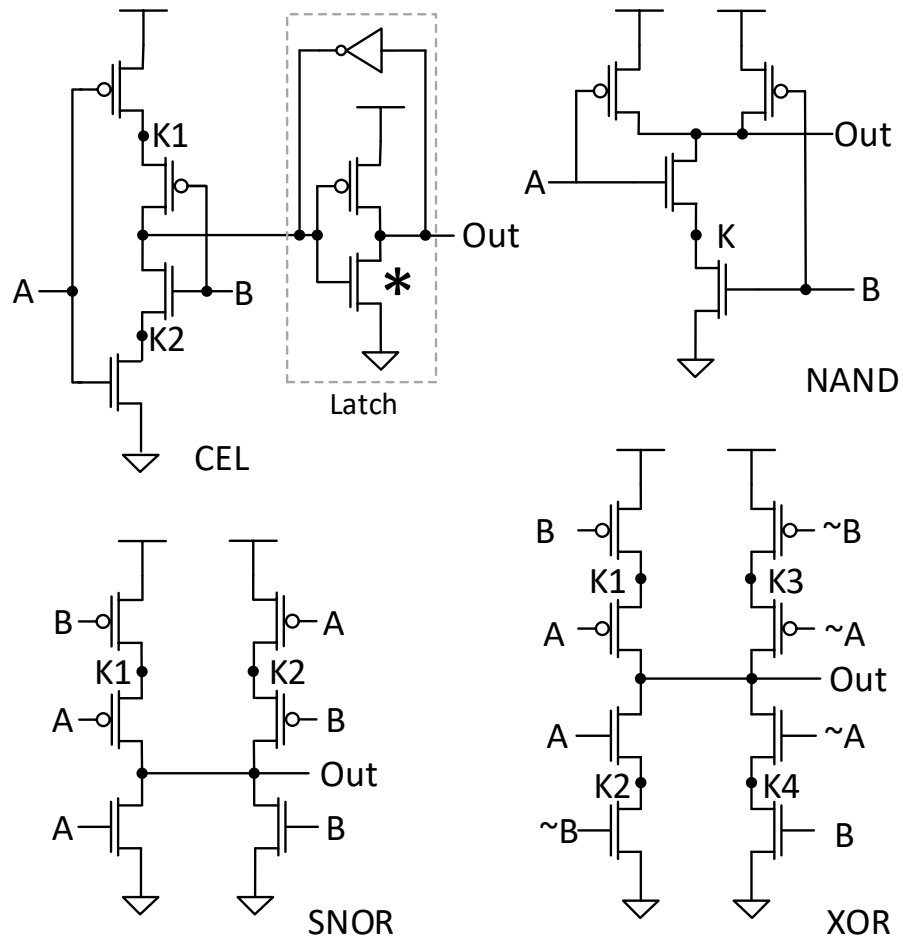


Figure 44: Four gate designs, with inputs A and B. All have internal K nodes that can cause drafting. These nodes are K1 in CEL (Micropipeline), K in the NAND (Click), both K1 and K2 in SNOR (GasP) depending on which input is asserted first, and K2 and K4 in XOR (Mousetrap). In CEL, the latch NMOS(*) completes the discharge path from K1.

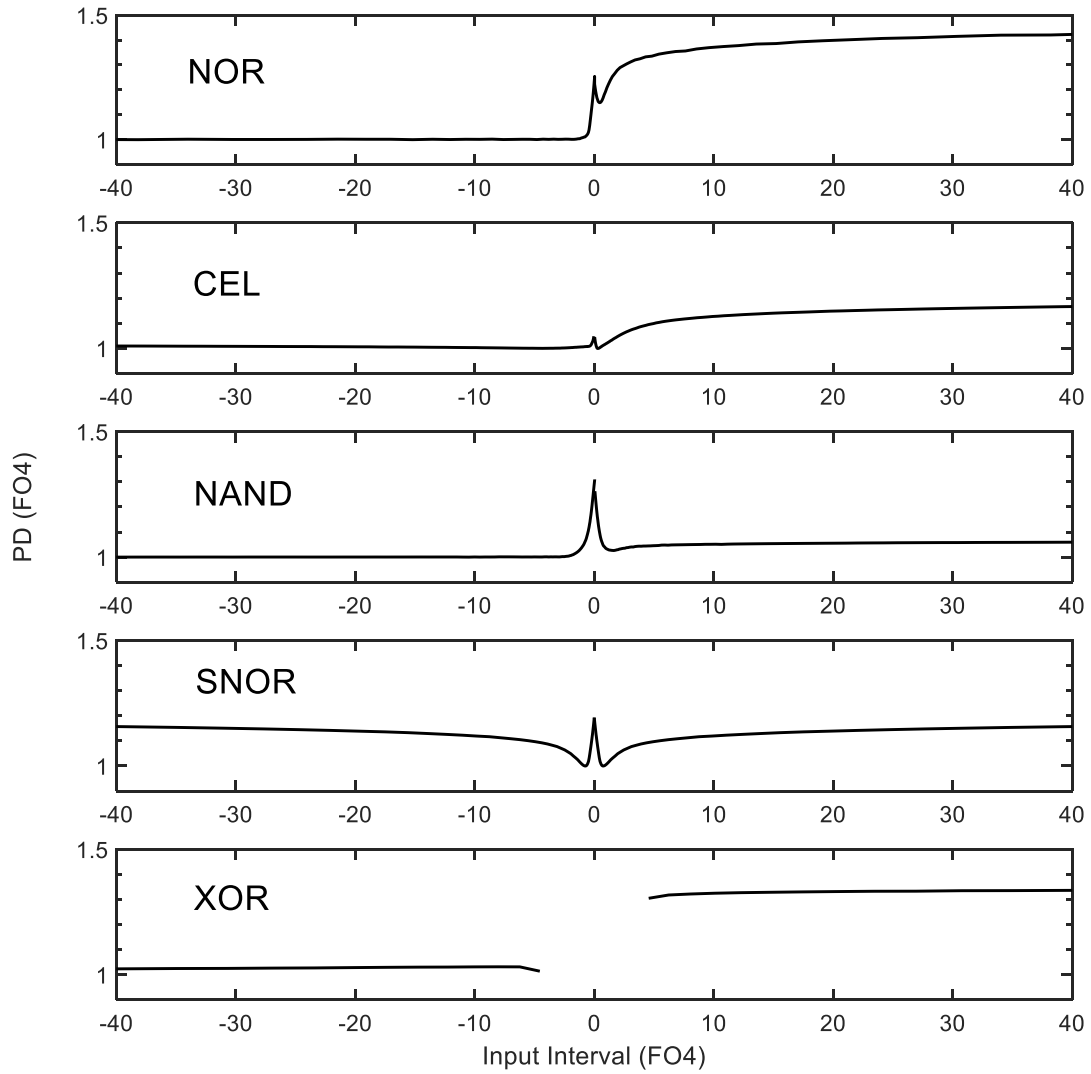


Figure 45: Propagation Delay (t_p) vs Input Interval for different decision gates. The gates were measured in SPICE in a FO4 buffered testbench. All show increasing propagation delay with increasing input interval which is the cause of drafting. The XOR, as used in Mousetrap, only operates with intervals greater than 4 FO4.

In Micropipeline there is an interesting case of K node decay without drafting. With a latch (Figure 46, right), the K1 node discharges to ground, the charge is lost and there is drafting. Without a latch (Figure 46, left), the CEL K1 node discharges to the output

node, net charge is not lost, propagation delay is fixed, and there is no drafting, in spite of slow K node decay.

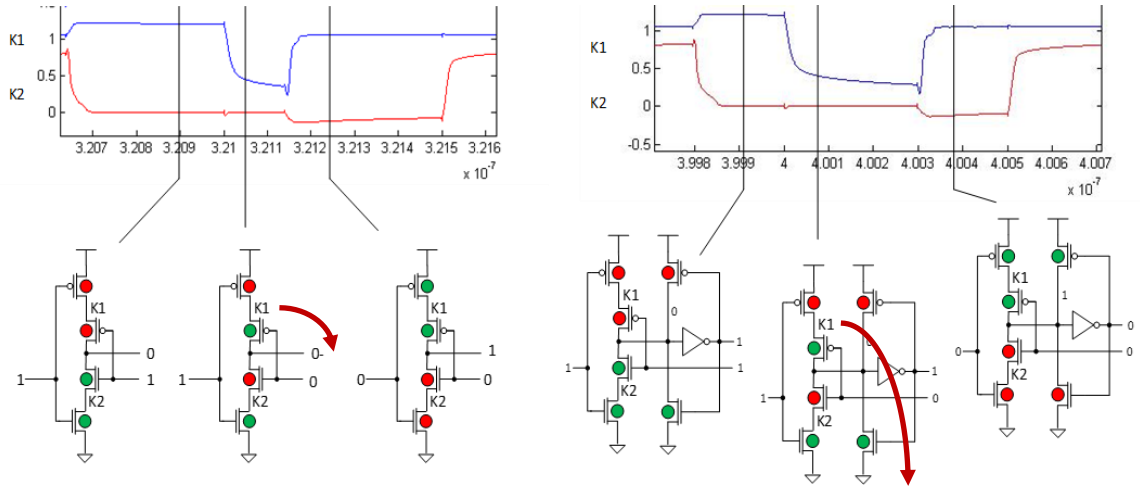


Figure 46: CEL operation in Micropipeline. Because of its high speed of operation, the CEL in Micropipeline will operate with (right) or without (left) an output latch. The typical K-node discharge is seen for K1 in the middle figures in both cases. Without a latch K1 (left) will discharge to the output node (short red arrow) and there is no loss of total charge and no drafting. With a latch (right) there is a discharge path to ground (long red arrow) for K1 which results in lost charge and drafting.

In Click [20], as realized in [21], the key decision gate is the NAND (Figure 47, red asterisk). A Click FIFO shows early pairing of tokens followed by drafting of intervals between pairs (Figure 47, right). The pairing is due to the different propagation delays of a bit 0 vs a bit 1 through the one-bit latch in the design during normal operation. This overwhelms the t_p variations of the NAND and creates longer t_p for odd intervals and shorter t_p for the even intervals. Eventually the pairs draft together from the effect of the NAND K node.

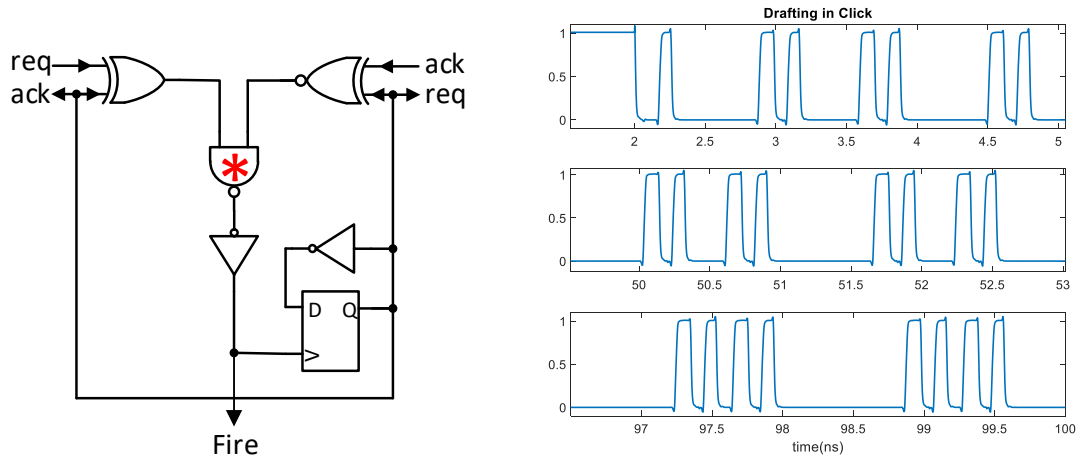


Figure 47: One Click stage from a test FIFO. The Click stage (left) decision gate NAND (asterisk) induces drafting. Fire signal traces (right) show early pairing of tokens caused by the one-bit latch, implemented by the D type flip-flop, followed by drafting of pairs of tokens.

Mousetrap [22], as realized in [3] is shown in Figure 48.

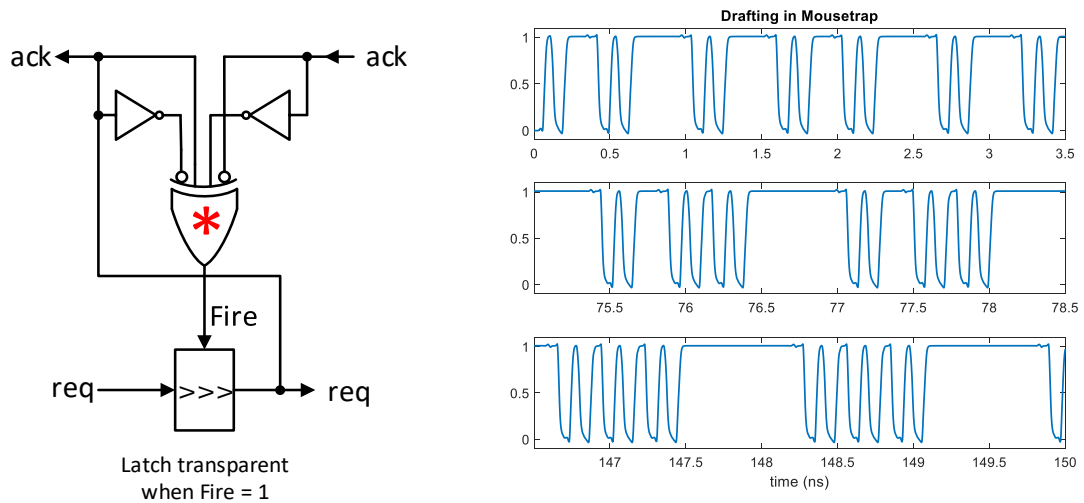


Figure 48: One stage of Mousetrap FIFO. The XOR decision gate (red asterisk) in the Mousetrap FIFO element (left) causes drafting. Note that the true and complement of the right acknowledge is reversed on the inputs to the XOR. Fire signal traces (right) show early pairing of tokens and eventual drafting of the pairs, as in Click.

Like Click, a Mousetrap FIFO also shows early pairing of tokens because it also uses a one-bit latch (Figure 48, right). Superimposed on this is a gradual drafting of the paired intervals due to an XOR K2 and K4 nodes. The XOR has 4 internal K nodes. Only two cause drafting in this implementation of Mousetrap (Figure 49).

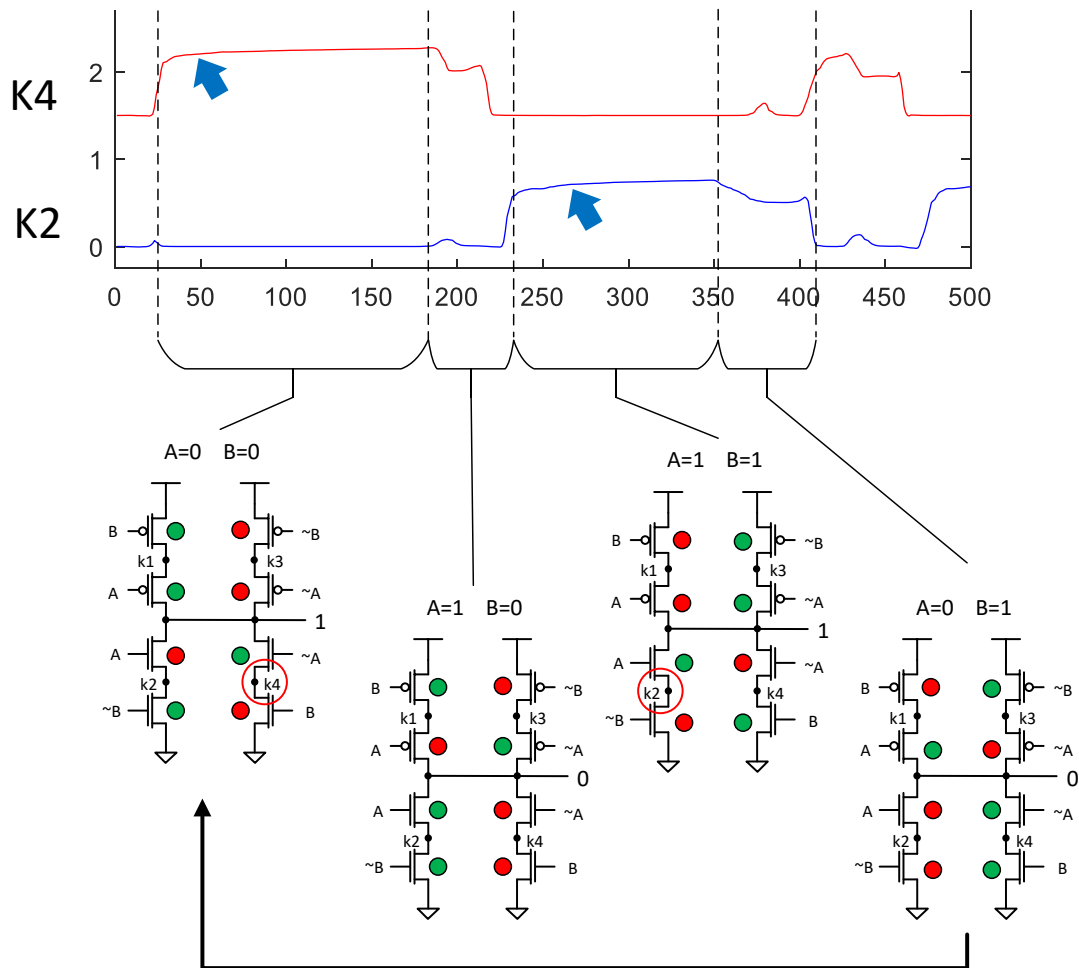


Figure 49: Phases of XOR operation in the Mousetrap implementation. Between odd tokens both A and B inputs are zero; between even tokens both A and B inputs are one. Between odd tokens the K4 node (red circle left) charges (blue arrow left). This affects the discharge time in the next phase which results in drafting. Between even tokens the K2 node (red circle right) node charges (blue arrow right) and again causes drafting.

A general principle can be derived from the decision gates examined here. Since all CMOS decision gates have at least one PMOS or NMOS stack in the design there will be one or more internal K nodes. When the timing of the two inputs to the gate arrive such that the inner (output-connected) transistor in the stack conducts before the outer (rail-connected) transistor the intervening K node will drift and there will be drafting. If the output node does not float between tokens there will be no drafting.

5.3 Additional K Nodes in GasP

If get-out-of-the-way (GOTW) keepers are included in the GasP design, the keepers are interrupted when the state-wires are being changed by an additional PMOS in series with the keeper drivers (Figure 50). Two additional internal K nodes added: one in the predecessor driver and another in the successor driver. In a sparse FIFO of this design the Pred K node is the one that drifts between tokens (Figure 51). The residual charge on that node slightly changes the rise time of charging the Pred state-wire and induces very slight drafting compared to the very strong decision-gate K-node drafting which is the main subject of this work. Hundreds of passes through the FIFO are required before there is any discernable drafting (Figure 52).

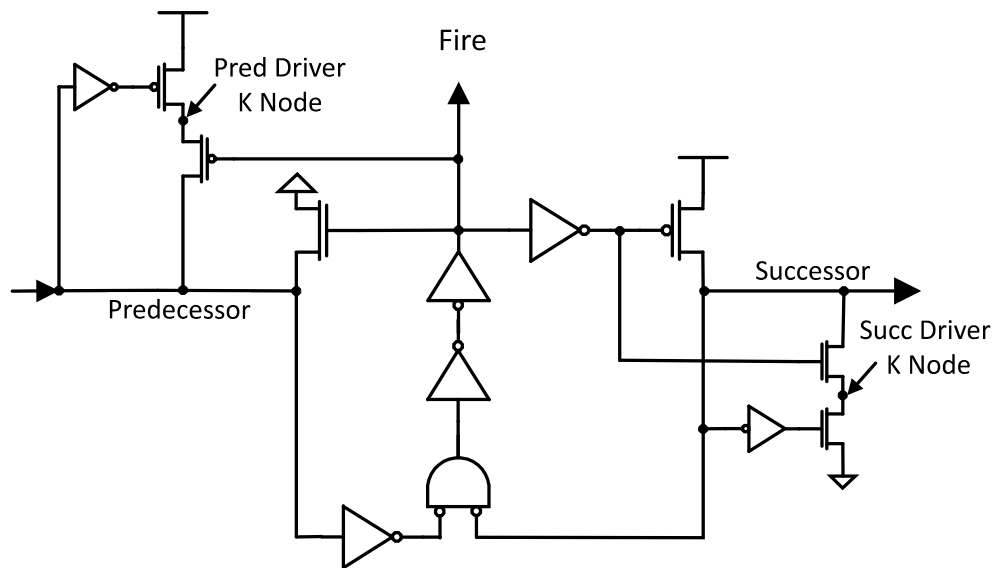


Figure 50: Two additional K nodes in GasP when using GOTW keepers.

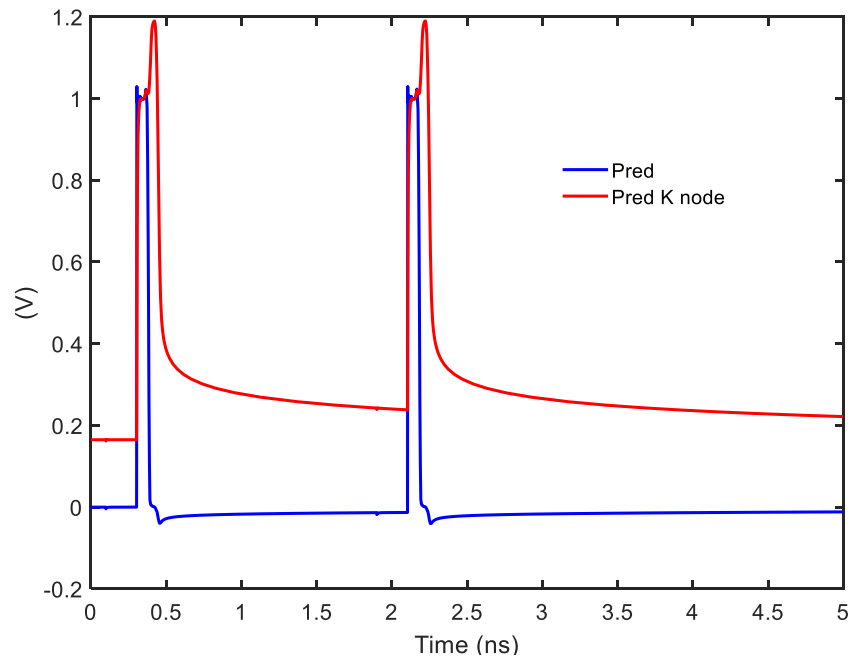


Figure 51: Get-out-of-the-way keeper Pred driver K node. The K node decays between tokens arriving on the Pred. The decay is exactly the same as the K node in the GasP NOR (see Figure 15).

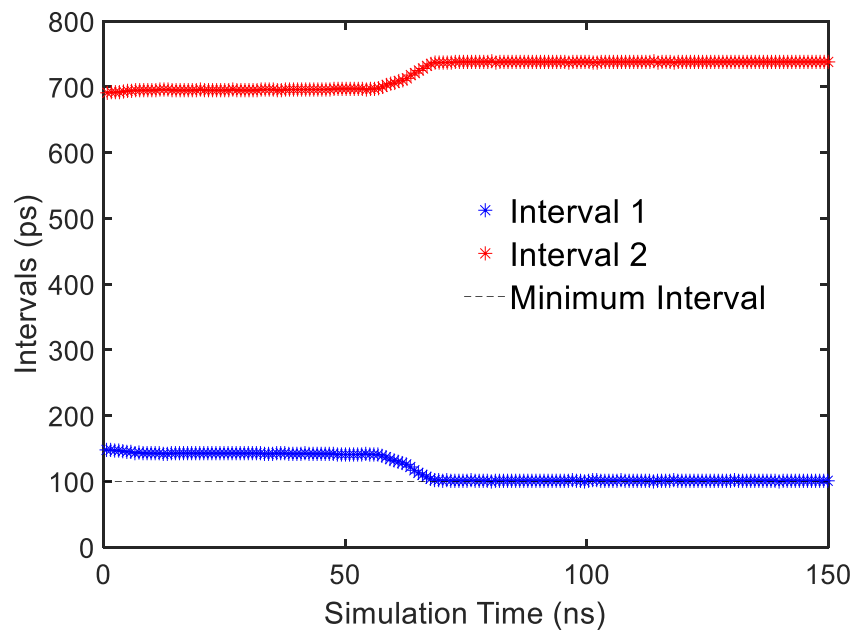


Figure 52: Very slow drafting due to get-out-of-the-way keepers in GasP. Hundreds of passes through the FIFO are required to accumulate any significant drafting.

Chapter 6 Summary and Future Work

The conventional explanation for drafting in self-timed circuits has been proven false. The state-wires are not incompletely charged or discharged during normal FIFO operation and all the input and output nodes of the gates are actively driven. The correct cause is the internal node in the decision gate. All decision gates have at least two inputs and at least one PMOS or NMOS stack. This stack contains an internal “K” node and it is the slow charging or discharging of this node between decision gate actions that causes drafting. The variable charge remaining on the K node when the decision gate asserts its output causes a variable propagation delay (t_p) through the gate. For drafting, t_p will increase as the spacing between events increases.

Now with drafting more fully understood it may be possible to use self-timed circuits for other applications: interval-based data/computation, spiking neural networks or in cryptography. Further work will characterize environmental and process variation of the drafting effect because, aside from mitigation, one can take advantage of manufacturing variability to create a physically un-cloneable devices (PUFs) [23]. The exact drafting behavior for any specific IC instance could provide unique identification or unique interval-based computation that could be used in cryptography.

6.1 Randomness and Unique Signatures

The drafting effect of any one stage is slight, so tokens need to move through many stages in a FIFO for there to be any observable change in intervals. These slight changes (~ps) are difficult to detect in a circuit operating at GHz speed. To amplify the effect of

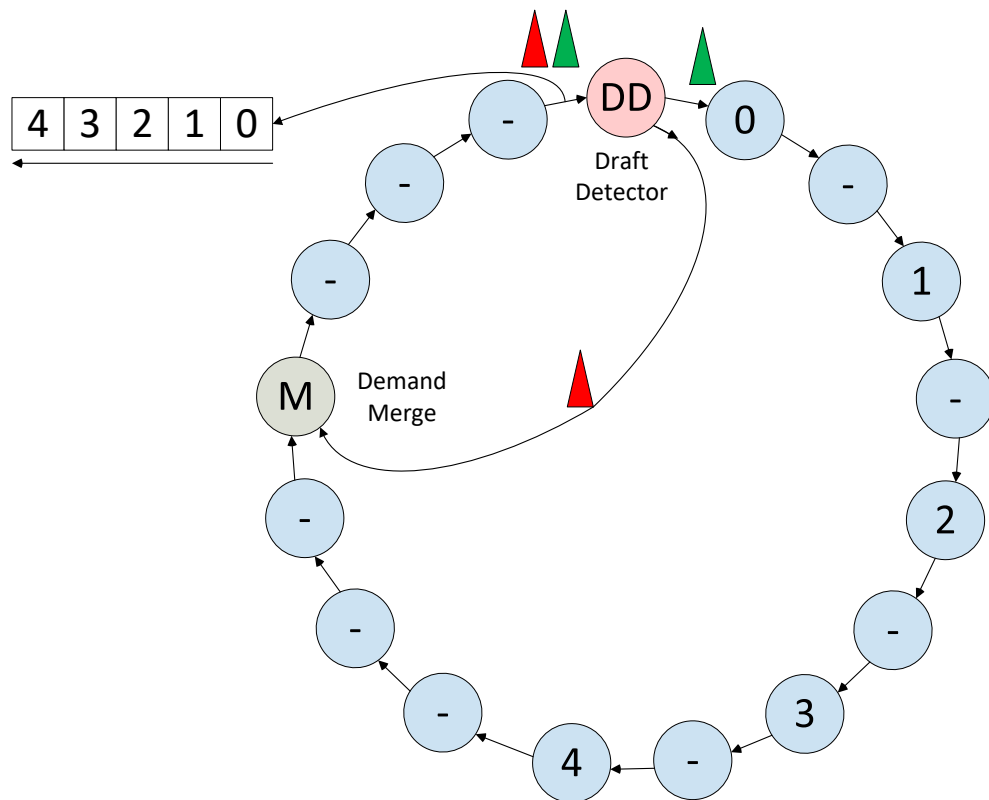


Figure 53: The shuffle circuit. This is a 17-stage circular FIFO with two special stages. One is the drafting detector and the other is the demand merge. The other 15 stages are ordinary GasP elements. 5 tokens are shown on the ring. Each token has bundled data, 0 through 5. The bundled data serves as a label to keep track of token identity. Empty GasP elements are indicated by “-“. With drafting the token order is shuffled so each token has a bundled identifier that can be recorded as the token passes.

The drafting detector circuit (Figure 54) uses two GasP stages; one outputs to the thru path and the other outputs to the bypass path. An existing thru token passes the bypass (red) and then the thru (blue) tapped state-wires. These state-wire-derived signals create a time window starting with bypass and ending with thru right behind the thru token. If a trailing token arrives in this window it will exit via the bypass route. If a token arrives well behind the previous token, outside the window, it will exit via the thru path. This circuit detects a range of closeness to the first token which is altered by drafting. When the drafting token gets too close to the token in front of it, it exits via the bypass route and gets reinserted.

The demand merge circuit (Figure 55) is a standard GasP design that combines tokens arriving on the thru and bypass routes to the thru route. It also contains two GasP stages but with arbitrated inputs and a common output. An arbiter assures that inputs are serviced first-come, first-serve, and none are lost in contention.

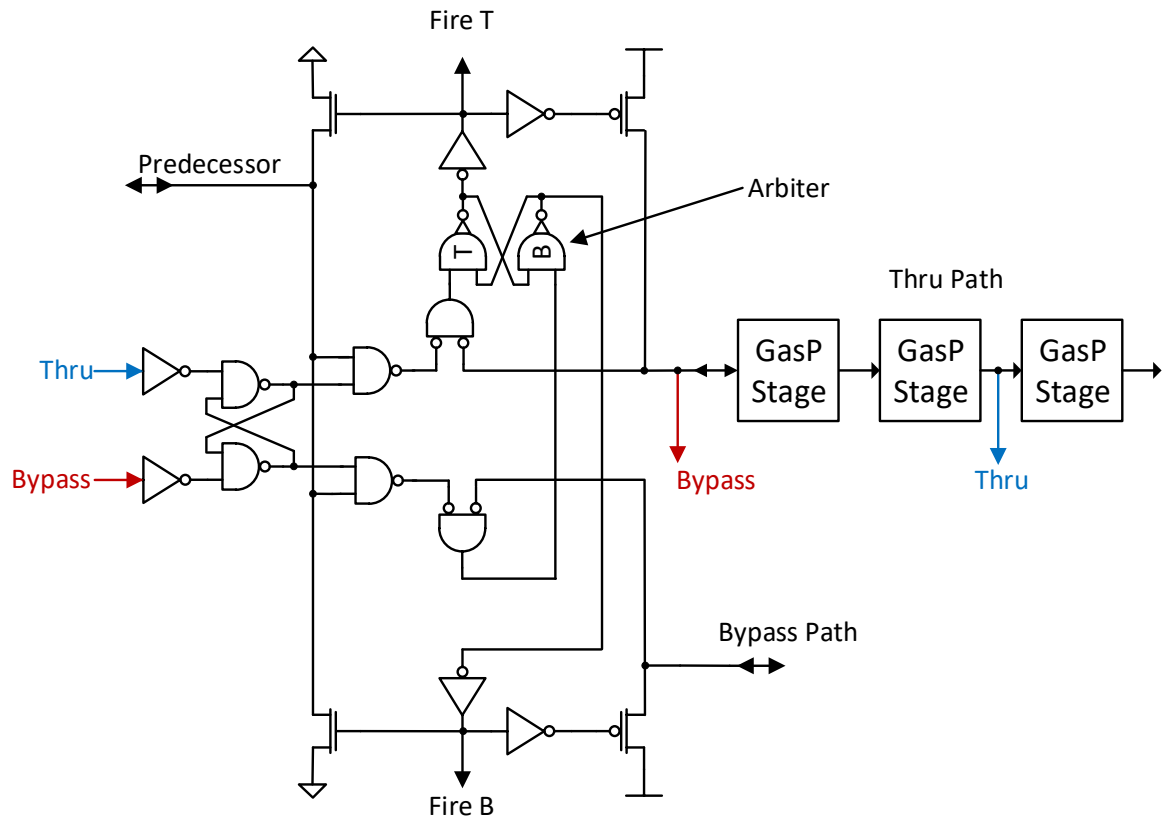


Figure 54: The drafting detector circuit. The first token passing through this circuit will exit via the thru path but triggers a time window. During this window the next token will exit via the bypass path. The two GasP circuits are arbitrated. Keepers and data latches are not shown for clarity.

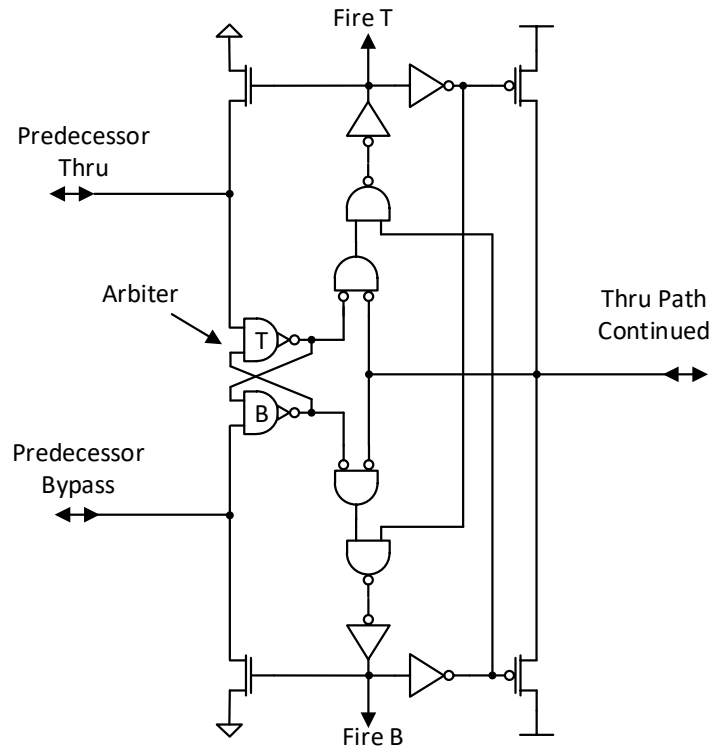


Figure 55: The GasP demand merge stage. Two GasP stages have arbitrated inputs and share a common output. Keepers and data latches are not shown for clarity.

A simulation of the shuffle circuit with 5 tokens, Figure 56, utilizing the special drafting detector and demand merge circuits as well as ordinary GasP stages was done using SPICE. Each stage has a data latch and the special stages have two latches. Bundled data is used to identify tokens since their sequence is altered. The token identifiers, 0 – 4, are recorded as they pass the drafting detector predecessor and are plotted graphically. Patterns are easier to see graphically than as a number stream. A 10 us simulation (28 hrs. in HSPICE) yields the stream in Figure 56. The stream progresses from lower left to upper right. Note that even after 10 us and 40,000 token passages, there is no repeating pattern. This appears to be at least quasi-random.

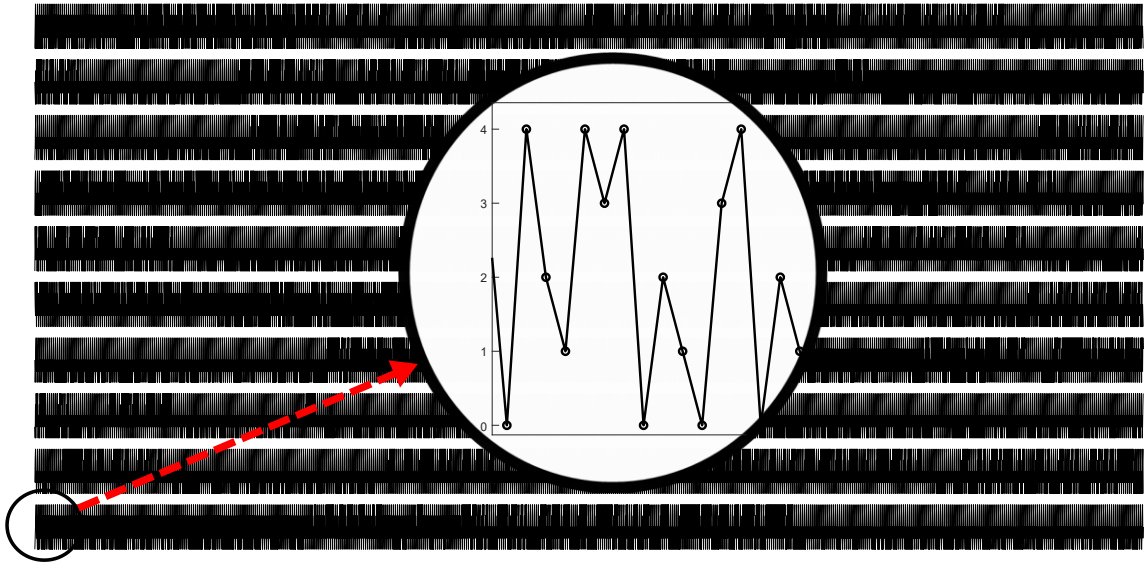


Figure 56: Shuffle circuit token stream. 40,000 tokens recorded over 10 us of simulation time in SPICE. The token number is plotted as a mini-graph. This makes it easier to see repeating patterns rather than viewing a sequence of digits. The stream progresses from lower left to upper right.

Six instances of process variation simulated by Monte Carlo SPICE simulation was generated and is shown in Figure 57. The last 2000 token passages for each instance is shown in the figure. Two instance have a short repeating sequence but the other four are unrepeating. All six are different which suggests that a similar shuffle circuit may be able to generate sequences that are instance-specific, which is useful for a PUF [23].

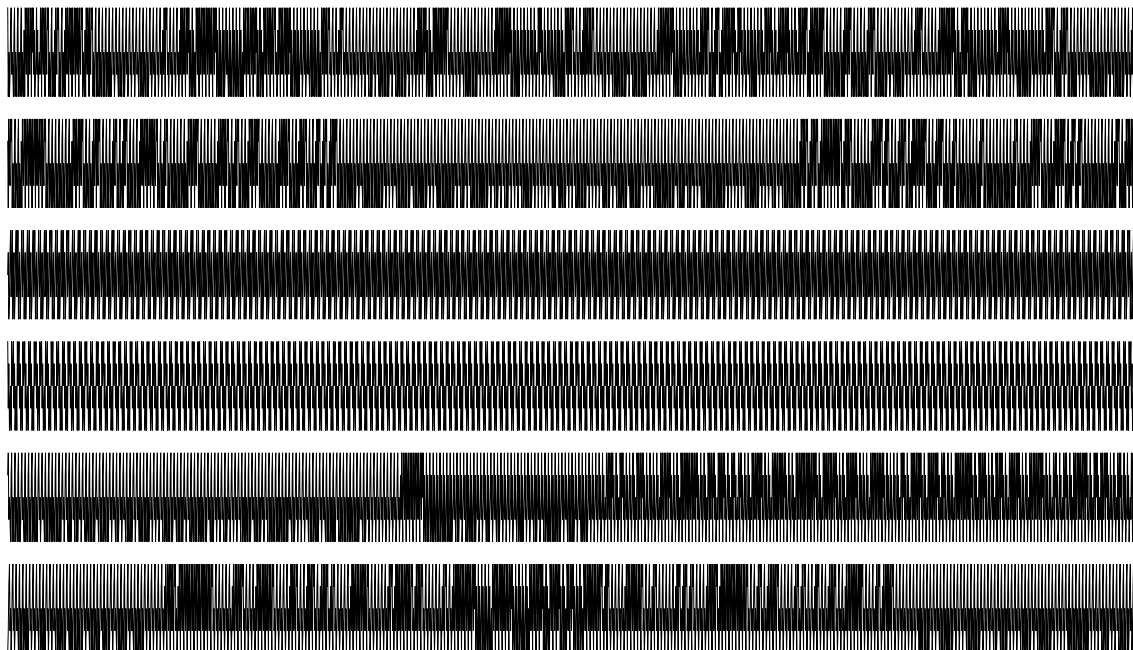


Figure 57: Six Monte Carlo simulations of the shuffle circuit. The last 2000 token passages of the 10 us runs displayed. Two instances show a repeating token pattern; the other four do not.

6.2 Data Encryption

If data is encoded as interval lengths perhaps drafting of the intervals could provide encryption on the sending end of a communications link consisting of two linear FIFOs if the drafting can be decrypted by anti-drafting at the receiving end (Figure 58). In view of the fact that we have shown in Chapter 4 that drafting and anti-drafting behavior can be controlled, it is useful to explore whether this control provides drafting reversal with sufficient fidelity.

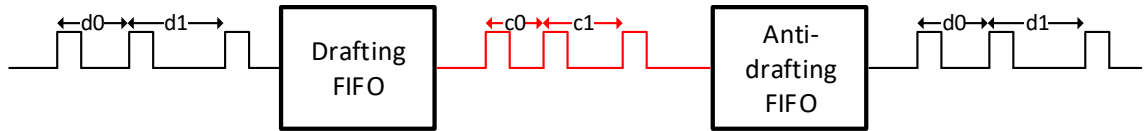


Figure 58: Concept of encrypting interval data by drafting. Data (d), encoded in intervals, is encrypted by drafting into cypher intervals (c). Anti-drafting decrypts and recovers the original data.

Anti-drafting can be achieved by inverting the drafting K-curve with a control circuit, Chapter 4 , so it might seem that this results in a perfect reversal of drafting. However, the following figures show that, in principle, drafting of more than one token in a token stream cannot be perfectly reversed by simply inverting the drafting K-curve. However, perfect reversal for two tokens (one interval) can be achieved by an anti-drafting curve with a slightly different shape than simple inversion. For more than 2 tokens (2 or more intervals) we show that reversal by passing tokens that have “drafted” into a FIFO with stages that all have a specific inverting “anti-drafting” K-curve is impossible

If the drafting curve is inverted and plotted with the anti-drafting curve, Figure 59, the figures will be less complex and easier to follow. Note that the curves are identical, and that the Y axis is now shortening and lengthening rather than t_{plh}

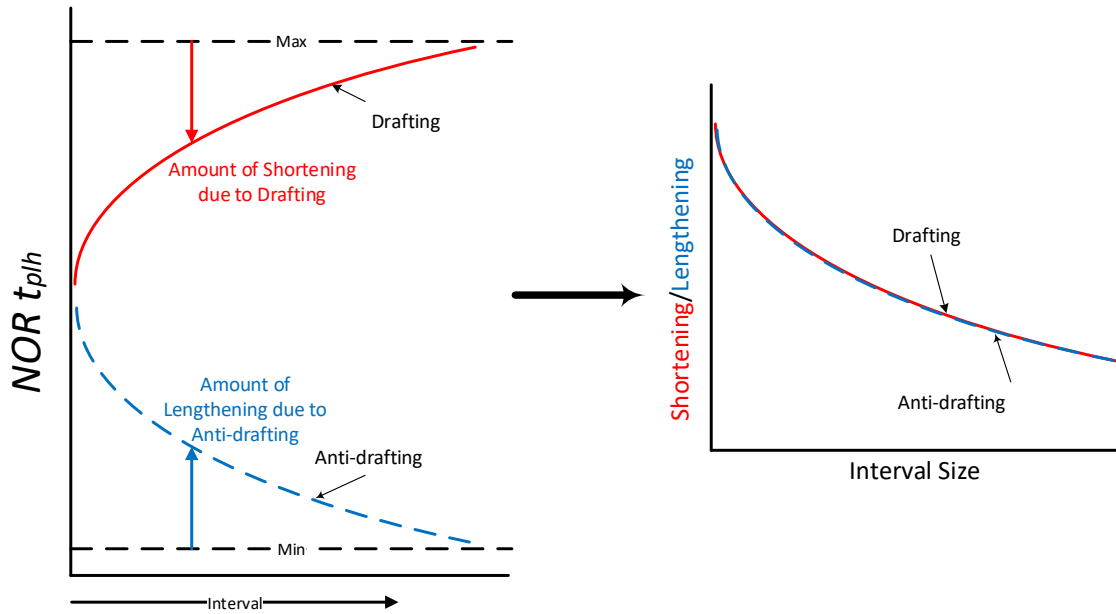


Figure 59: Inverting the drafting curve to simplify the diagram.

If one interval is drafted and then anti-drafted in one FIFO stage by identical curves there is an error (Figure 60). Logically this follows since if one increases something by 20% and then shrinks the result by 20% you will not recover the original. If the anti-drafting curve is adjusted, then complete reversal of drafting will occur with one interval (Figure 61). The adjustment is

$$ad(I) = d(I + d(I)) \quad (19)$$

where d is the drafting function and ad is the anti-drafting function. This can be applied to any drafting curve function.

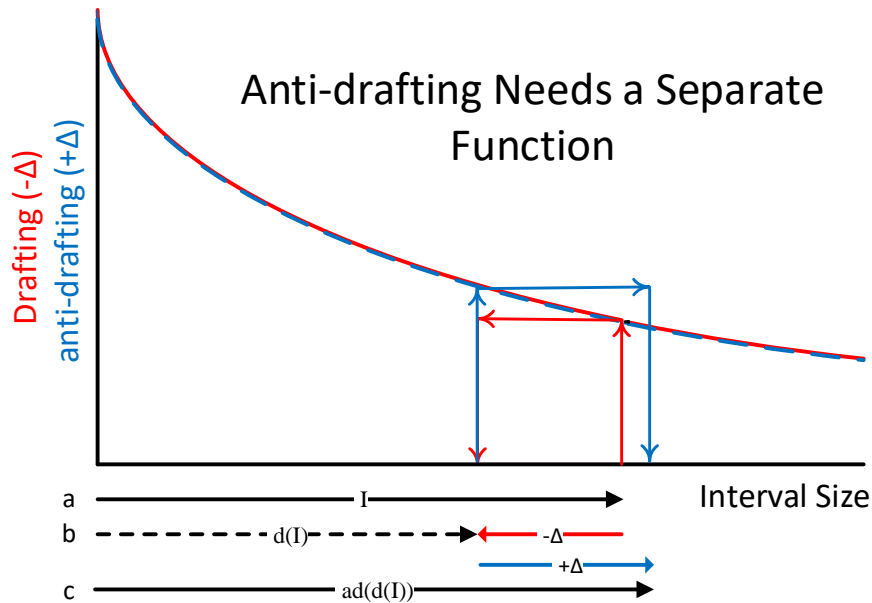


Figure 60: An inverted anti-drafting curve will not perfectly reverse drafting. Given a starting interval I (a) it will be shortened by $-\Delta$ by the drafting curve (b). The result, $d(I)$ will then be lengthened by $+\Delta$ from the anti-drafting curve (c). The result is not equal to the original interval.

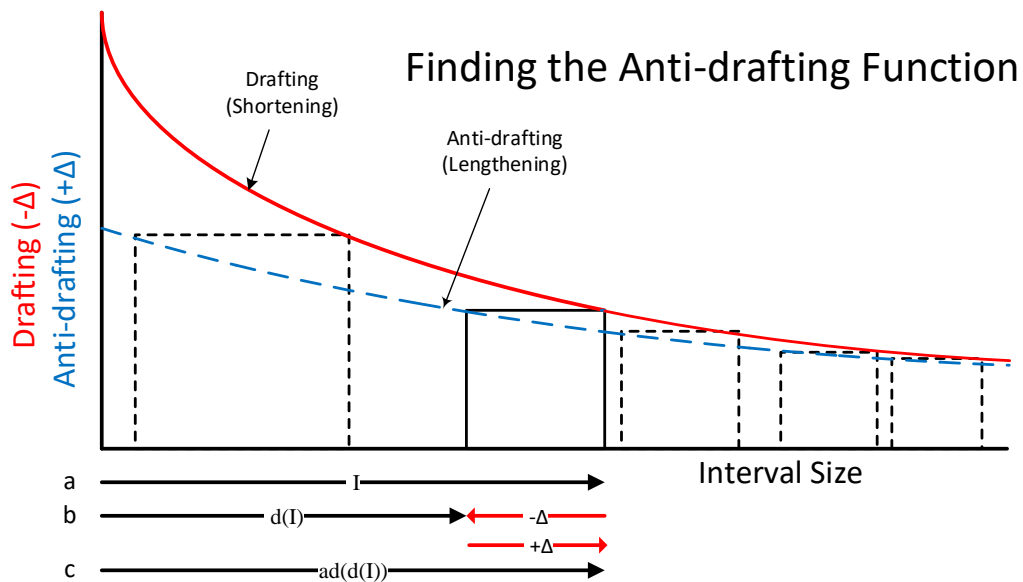


Figure 61: Construction of a perfect anti-drafting curve. The flipped anti-drafting curve has a variable offset from the drafting curve on the interval size axis. Starting with interval I (a) it will be shortened by $-\Delta$ by the red drafting curve. The shortened interval $d(I)$ will be lengthened by $+\Delta$ using the adjusted blue anti-drafting curve (c). This construction provides perfect reversal. With

more than one interval (more than 2 tokens on a linear FIFO) the preceding approach will not work. The first interval will be perfectly reversed but the trailing intervals will show error. This can be shown geometrically for two intervals (3 tokens on a linear FIFO). Figure 62 shows the drafting of the two intervals and Figure 63 shows the anti-drafting of the drafted intervals. The key is the change in interval 2 that must occur when interval 1 is shortened (Figure 62, c). Interval 2 is no longer the original and the *change is not stored anywhere in the circuit*. The results of drafting and then anti-drafting of two intervals is summarized in Figure 64.

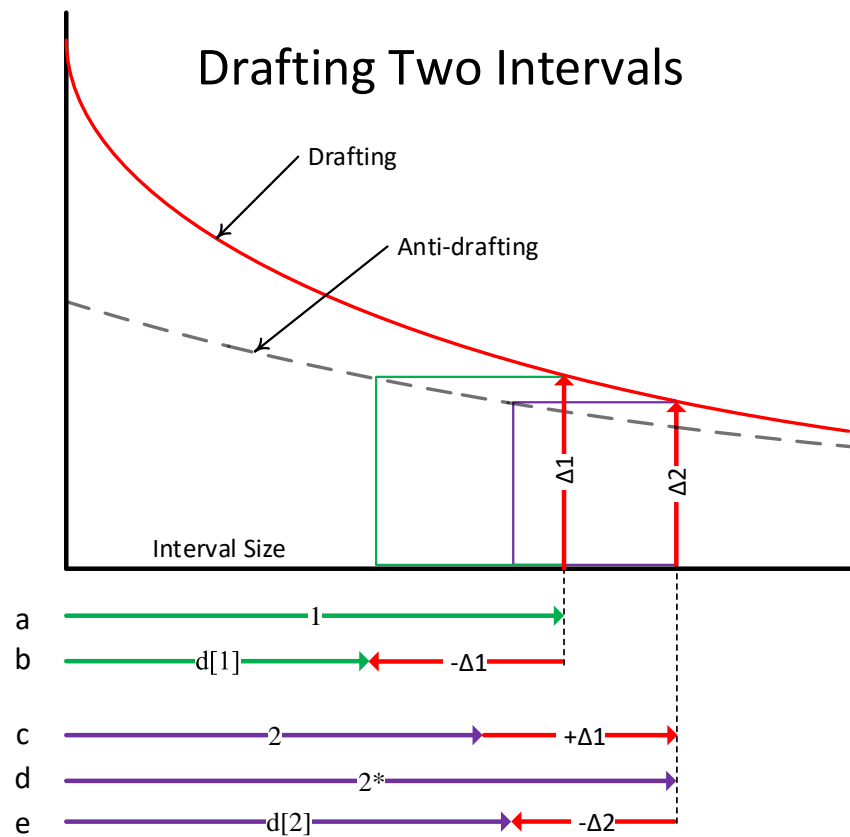


Figure 62: Drafting two intervals. The first interval (a) is shortened by $-\Delta$ (b). The next interval, 2 will be lengthened (c) by the same amount to 2^* (d). The lengthened second interval then shortens by drafting based on interval 2^* not 2 (e). This happens for all trailing intervals.

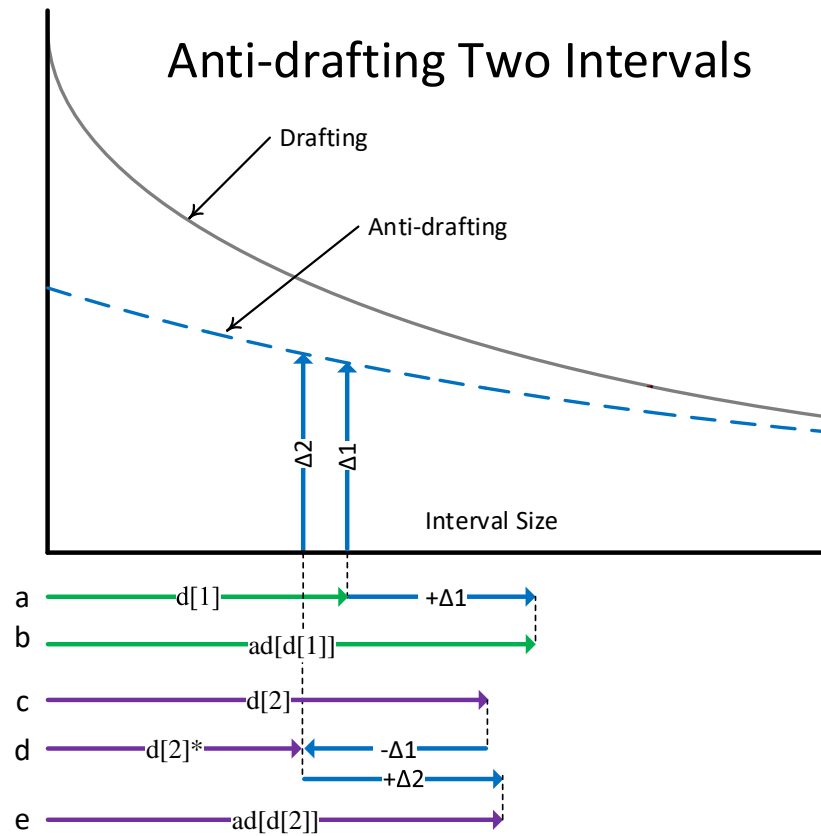


Figure 63: Two succeeding intervals undergoing anti-drafting. The drafted intervals $d[1]$ and $d[2]$ are taken from Figure 62. Interval $d[1]$ is anti-drafted by $+\Delta 1$ to yield $ad[d[1]]$ in (a). The next interval, $d[2]$ (c) will be shortened by the same amount (d) and the modified interval $d[2]^*$ will be anti-drafted by $+\Delta 2$.

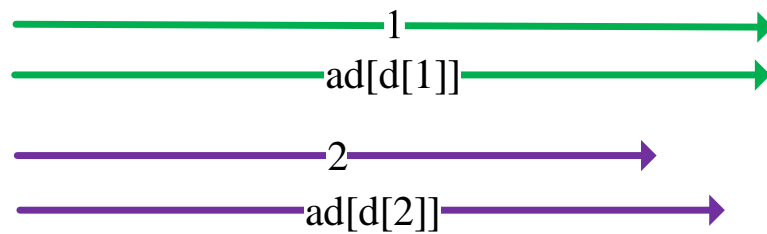


Figure 64: Results after attempted reversal of drafting for two intervals. The first interval is accurately reversed but the second interval is not. The error continues into the third and subsequent intervals.

This reversal error creates a constraint on using drafting and its reverse in practice. The error diminishes as the intervals get larger but so does the amount of change in the intervals and the degree of encryption. Further work could establish the amount of acceptable error that would make drafting for encryption practical.

References

- [1] I. Sutherland and S. Fairbanks, "GasP: A minimal FIFO control," in *Asynchronous Circuits and Systems, 2001. ASYNC 2001. Seventh International Symposium on*, 2001, pp. 46-53.
- [2] K. E. Molnar, I. W. Jones, W. S. Coates, and J. K. Lexau, "A FIFO ring performance experiment," in *Advanced Research in Asynchronous Circuits and Systems, 1997. Proceedings., Third International Symposium on*, 1997, pp. 279-289.
- [3] M. Roncken, S. M. Gilla, H. Park, N. Jamadagni, C. Cowan, and I. Sutherland, "Naturalized communication and testing," in *Asynchronous Circuits and Systems (ASYNC), 2015 21st IEEE International Symposium on*, 2015, pp. 77-84.
- [4] S. M. Gilla, M. Roncken, and I. Sutherland, "Long-range GasP with charge relaxation," in *Asynchronous Circuits and Systems (ASYNC), 2010 IEEE Symposium on*, 2010, pp. 185-195.
- [5] S. Ghosh-Dastidar and H. Adeli, "Spiking neural networks," *International journal of neural systems*, vol. 19, pp. 295-308, 2009.
- [6] J. Kalisz, "Review of methods for time interval measurements with picosecond resolution," *Metrologia*, vol. 41, p. 17, 2003.
- [7] A. Winstanley and M. Greenstreet, "Temporal Properties of Self-Timed Rings," Berlin, Heidelberg, 2001, pp. 140-154.
- [8] A. J. Winstanley, A. Garivier, and M. R. Greenstreet, "An event spacing experiment," in *Asynchronous Circuits and Systems, 2002. Proceedings. Eighth International Symposium on*, 2002, pp. 47-56.
- [9] V. Chandramouli and K. A. Sakallah, "Modeling the effects of temporal proximity of input transitions on gate propagation delay and transition time," in *Design Automation Conference Proceedings 1996, 33rd*, 1996, pp. 617-622.
- [10] J. C. Ebergen, S. Fairbanks, and I. E. Sutherland, "Predicting performance of micropipelines using Charlie diagrams," in *Advanced Research in Asynchronous Circuits and Systems, 1998. Proceedings. 1998 Fourth International Symposium on*, 1998, pp. 238-246.
- [11] J. Hamon, L. Fesquet, B. Miscopein, and M. Renaudin, "Constrained Asynchronous Ring Structures for Robust Digital Oscillators," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 17, pp. 907-919, 2009.
- [12] A. Cherkaoui, V. Fischer, A. Aubert, and L. Fesquet, "A Self-Timed Ring Based True Random Number Generator," in *Asynchronous Circuits and Systems (ASYNC), 2013 IEEE 19th International Symposium on*, 2013, pp. 99-106.
- [13] J. Hamon, L. Fesquet, B. Miscopein, M. Renaudin, and I. C. Soc, *High-level time-accurate model for the design of self-timed ring oscillators*. Los Alamitos: Ieee Computer Soc, 2008.
- [14] E. Yahya, O. Elissati, H. Zakaria, L. Fesquet, and M. Renaudin, "Programmable/Stopable Oscillator Based on Self-Timed Rings," in *2009 15th IEEE Symposium on Asynchronous Circuits and Systems*, 2009, pp. 3-12.

- [15] V. Zebilis and C. P. Sotiriou, "Controlling event spacing in self-timed rings," in *Asynchronous Circuits and Systems, 2005. ASYNC 2005. Proceedings. 11th IEEE International Symposium on*, 2005, pp. 109-115.
- [16] S. Fairbanks and S. Moore, "Analog micropipeline rings for high precision timing," in *Asynchronous Circuits and Systems, 2004. Proceedings. 10th International Symposium on*, 2004, pp. 41-50.
- [17] S. M. Fairbanks, "High precision timing using self-timed circuits," Citeseer, 2005.
- [18] A. Chatzigeorgiou, S. Nikolaidis, and I. Tsoukalas, "A modeling technique for CMOS gates," *Trans. Comp.-Aided Des. Integ. Cir. Sys.*, vol. 18, pp. 557-575, 2006.
- [19] L.-C. Chen, S. K. Gupta, and M. A. Breuer, "A new gate delay model for simultaneous switching and its applications," presented at the Proceedings of the 38th annual Design Automation Conference, Las Vegas, Nevada, USA, 2001.
- [20] A. Peeters, F. Te Beest, M. De Wit, and W. Mallon, "Click elements: An implementation style for data-driven compilation," in *Asynchronous Circuits and Systems (ASYNC), 2010 IEEE Symposium on*, 2010, pp. 3-14.
- [21] Y. Liu, H. Chen, D. Wang, and A. He, "An asynchronous loop structure based on the click element," in *Electron Devices and Solid-State Circuits (EDSSC), 2017 International Conference on*, 2017, pp. 1-2.
- [22] M. Singh and S. M. Nowick, "MOUSETRAP: Ultra-high-speed transition-signaling asynchronous pipelines," in *iccd*, 2001, p. 0009.
- [23] C.-H. Chang, Y. Zheng, and L. Zhang, "A retrospective and a look forward: Fifteen years of physical unclonable function advancement," *IEEE Circuits and Systems Magazine*, vol. 17, pp. 32-62, 2017.